НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО

Факультет інформатики та обчислювальної техніки
(назва факультету, інституту)

Кафедра автоматизованих систем обробки інформації і управління
(назва кафедри)

"На правах рукопису"
УДК *519.854.2*

«До захисту допущено»
Завідувач кафедри

_____ О.А.Павлов
(підпис) (ініціали, прізвище)

“____” _____ 20*18* р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

на здобуття ступеня магістра

за спеціальністю *122 Комп'ютерні науки та інформаційні технології*
(код та назва спеціальності)

спеціалізацією *Інформаційні управляючі системи та технології*
(код та назва спеціалізації)

на тему: *Задача ефективного розподілу навантаження між електростанціями*

Виконав: студент *VI* курсу групи ІС-63м
(шифр групи)

Бабич Світлана Олександрівна _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник *проф., д.т.н., с.н.с. Гуляницький Л. Ф.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант *к.т.н., доц. Жданова О.Г.* _____
(науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018

РЕФЕРАТ

Магістерська дисертація: 107 с., 22 рис., 29 табл., 7 додатків, 77 джерел.

Актуальність. На сьогоднішній день важко уявити наше життя без пристроїв, що споживають електроенергію. У порівнянні із XX століттям споживання електроенергії зросло більше, ніж у 100 разів. Розташування електростанцій залежить від можливості постачання енергоносіїв, через що вони, зазвичай, розмішуються з огляду на можливість їх безперешкодного та простого постачання. Електростанції часто розташовуються досить далеко від основних споживачів електроенергії, тому частина електроенергії втрачається під час транспортування. Але значно більш важливою проблемою є неможливість економічно ефективного зберігання виробленої електроенергії, що спричинює значні матеріальні втрати при неправильному плануванні економічного навантаження між електростанціями.

Розроблено ряд методів та алгоритмів для заходження ефективного розподілу навантаження між електростанціями в електромережі, але кожен із них має певні недоліки, що дозволяє його ефективне використання лише для певної підможини задачі економічного розподілу навантаження та динамічного розподілу навантаження. З огляду на це дана задача є актуальною сьогодні. Її особливість полягає у нелінійності функцій генерації електроенергії, що у значній мірі ускладнює розробку ефективного алгоритму.

Мета дослідження – розробка математичного апарату, застосування якого спрямоване на зменшення витрат палива на виробництво електроенергії

Для досягнення поставленої мети необхідно виконати наступні **завдання**:

- виконати огляд існуючих методів розв’язування поставленої задачі;
- формалізувати задачі економічного розподілу навантаження та динамічного розподілу навантаження із врахуванням обмежень, які виникають під час виробництва електроенергії;
- розробити алгоритми розв’язування поставлених задач;
- здійснити програмну реалізацію запропонованих алгоритмів;
- провести експериментальні дослідження розроблених алгоритмів;

- виконати аналіз отриманих результатів.

Об’єкт дослідження – процес розподілу навантаження між електростанціями для замкнутої енергетичної системи. **Предмет** дослідження – методи ефективного розподілу навантаження між електростанціями.

Наукова новизна отриманих результатів – розроблено оригінальний алгоритм вовчої зграї для розв’язування задач економічного розподілу навантаження та динамічного розподілу навантаження. Наведено спосіб зведення отриманого розв’язку до допустимого для поставлених задач.

Публікації. Матеріали роботи опубліковані в статті в міжнародному науковому журналі «Науковий огляд» [1,2].

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалась у філії кафедри автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках науково-дослідної теми Інституту кібернетики ім. В. М. Глушкова НАН України: «Розробити математичний апарат, орієнтований на створення інтелектуальних інформаційних технологій розв’язування проблем комбінаторної оптимізації та інформаційної безпеки» (шифр теми: ВФ.180.11).

ЕКОНОМІЧНИЙ РОЗПОДІЛ НАВАНТАЖЕННЯ, ДИНАМІЧНИЙ РОЗПОДІЛ НАВАНТАЖЕННЯ, АЛГОРИТМ ВОВЧОЇ ЗГРАЇ, ОПТИМІЗАЦІЯ РОЄМ ЧАСТИНОК, ГЕНЕТИЧНИЙ АЛГОРИТМ, РОЙОВІ АЛГОРИТМИ, ЕЛЕКТРОЕНЕРГЕТИКА

ABSTRACT

Master’s thesis: 107 pages, 22 figures, 29 tables, 7 appendix, 77 references.

Relevance. Today, it's impossible to imagine our life without appliances that consume electricity. Compared with the 20th century, electricity consumption has grown more than 100 times. The location of power plants depends on the ability to supply energy, which is why they are usually hugged with the possibility of their unobstructed and simple supply. Because of this, power stations are often located far enough away from the main consumers of electricity, which is why part of the electricity is lost during transportation. But a much more important problem is the impossibility of economically efficient storage of produced electricity, which causes significant material losses in case of improper planning of economic load between power plants.

A lot of methods and algorithms have been developed for the establishment of a cost-effective distribution of the load between the power plants in the grid, but each of them has certain disadvantages, which allows its effective use only for a certain part of the task of economic load dispatch and dynamic load dispatch. Given this, the task is relevant today. Its feature is the nonlinearity of the functions of generating electricity, which greatly complicates the development of an efficient algorithm.

Purpose and objectives of the study is reduction of fuel consumption for electricity production due to efficient allocation of load between power plants.

To achieve the goal must perform the following **tasks**:

- to perform an overview of existing methods of solving the task;
- to formalize the task of economic load dispatch and dynamic load dispatch;
- implement algorithms for the tasks;
- develop a software implementation of a modified algorithm;
- to conduct experimental research of existing and developed algorithms;
- to perform the analysis of the results.

The object of the study is the process of distributing the load between the power plants for a closed power system. **Subject** of the study is methods of effective distribution of load between power plants.

Scientific novelty of the results. The modification of the grey wolf optimizer for solving the problems of economic load distribution and dynamic load distribution has been developed. The method of transforming received solution of the defined problem to the possible one is provided.

Publications. Publications Materials of the work are published in the article in the international scientific journal “Scientific Review” [1,2].

Relationship of work with scientific programs, plans, themes. Connection of the thesis with scientific programs, plans, topics. The thesis was written at the branch of The Department of Department of Computer-aided management and data processing systems of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute” at the V. M. Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine under the topic “Develop a mathematical apparatus aimed at creating of intellectual information technologies for solving combinatorial optimization and information security problems” (topic’s index is ВФ.180.11).

ECONOMIC LOAD DISPATCH, DYNAMIC LOAD DISPATCH, GREY WOLF OPTIMIZER, PARTICLE SWARM OPTIMIZATION, GENETIC ALGORITHM, SWARM INTELLIGENCE, ELECTRIC POWER INDUSTRY

ЗМІСТ

Вступ 9

1 Огляд методів розв’язання задач ефективного розподілу навантаженя 11

1.1 Методи лінійного програмування 12

1.2 Методи нелінійного програмування 13

1.3 Методи стохастичного програмування 14

1.4 Методи мета евристичної оптимізації 15

1.5 Гібридні технології. 23

Висновки до розділу 26

2 Постановка та формалізація задачі розподілу навантаження 27

2.1 Змістова постановка задачі 27

2.2 Математична модель задачі 28

2.2.1 Задача економічного розподілу навантаження 28

2.2.2 Задача економічного розподілу навантаження без врахування втрат 32

2.2.3 Задача економічного розподілу навантаження із точковим навантаженням клапанів 33

2.2.4 Задача динамічного розподілу навантаження 34

2.2.5 Зведення задачі ДРН до задачі із неперервним простором визначення 36

Висновки до розділу 39

3 Алгоритми розв’язування задачі 40

3.1 Оптимізація роєм частинок 40

3.2 Оптимізація вовчою зграєю 43

3.3 Конкретизація алгоритму вовчої зграї 47

3.4 Генетичний алгоритм 49

3.5 Алгоритм зведення розв’язку до допустимого 55
Висновки до розділу 63
4 Опис програмного забезпечення 64
4.1 Призначення та функції 64
4.2 Середовище розробки та використані технології 64
4.3 Вимоги до технічного забезпечення 67
4.4 Опис програмної реалізації 68
4.5 Опис вхідних та вихідних даних 70
Висновки до розділу 72
5 Дослідження ефективності алгоритмів 74
5.1 Процес проведення експериментів 74
5.2 Порівняння ефективності роботи алгоритмів 80
5.3 Дослідження алгоритму зведення розв’язку до допустимого 88
Висновки до розділу 90
Висновки 91
Перелік посилань 93
Плакат 1 Результати експериментальних досліджень 101
Плакат 2 Діаграма класів 102
Плакат 3 Діаграма пакетів 103
Плакат 4 Діаграма послідовності 104
Плакат 5 Блок-схеми розроблених алгоритмів: ОРЧ, АВЗ та модифікація АВЗ ... 105
Плакат 6 Блок-схеми процедур зведення розв’язку до допустимого 106
Плакат 7 Екранні форми 107

ВСТУП

Одним із найважливіших завдань планування та експлуатації системи електропостачання є ефективне планування роботи всіх генераторів у системі для задоволення необхідного попиту. Задачі економічного розподілу навантаження та динамічного розподілу навантаження пов’язані з ефективним розподілом навантаження між генераторами електростанцій в електромережі, що мінімізує загальну вартість палива аби задовольнити попит на навантаження та операційні обмеження. Даний вид задач відіграє важливу роль у плануванні та в управлінні сучасними енергосистемами.

В останні кілька десятиліть для розв’язування задач економічного розподілу навантаження було застосовано низку підходів, таких як математичне програмування (градієнтний метод), лінійне програмування, динамічне програмування, метод невизначених множників Лагранжа, метод гілок та меж та інших. Але, з огляду на особливості форми області значень цільової функції оптимізаційної задачі, зазвичай вони затримуються в локальних оптимумах цільової функції. Крім того, вхідні та вихідні характеристики сучасних генераторів за своєю природою є нелінійними з різних причин, серед яких ефект різнорідного палива, нерівномірного навантаження вузлів (клапанів), обмеження швидкості зміни потужності та інші. Ці характеристики і призводять до виникнення цілого ряду локальних оптимумів, які створюють проблеми із знаходженням глобального оптимуму. Також для розв’язування задачі використовують стохастичні метаевристичні алгоритми, такі як диференціальна еволюція та генетичний алгоритм. Проте на даний момент існує недостатня кількість досліджень присвячених задачі динамічного розподілу навантаження, якщо порівнювати із іншими задачами в області електроенергетичної оптимізації включаючи задачу економічного розподілу навантаження.

При детальному дослідженні наукових робіт було виділено три алгоритми: генетичний, оптимізації роєм частинок та вовчої зграї. Дані алгоритми були

реалізовані для перевірки ефективності модифікації алгоритму. Модифікований алгоритм базується на основі алгоритму вовчої зграї. Для розв’язування поставлених задач було прийнято рішення виконати перехід від задачі із обмеженнями, що характеризують фізичні та експлуатаційні характеристики електростанцій, до задачі без обмежень із використанням функцій штрафів. Також важливим кроком є етап «доопрацювання» отриманого розв’язок після завершення роботи основного алгоритму. На цьому кроці виконується зведення розв’язку до допустимого.

1 ОГЛЯД МЕТОДІВ РОЗВ'ЯЗАННЯ ЗАДАЧ ЕФЕКТИВНОГО РОЗПОДІЛУ НАВАНТАЖЕНЯ

В основі парадигми минулого XX століття, століття інтенсивної індустріалізації та науково-технічної революції, була закладена концепція максимального зосередження матеріальних, фінансових, енергетичних та людських ресурсів для виробництва промислових, сільськогосподарських чи інших видів продукції. З того часу, основним завданням електроенергетики було досягнення ефективності електропостачання, що вимагало високої концентрації промислових та енергетичних потужностей і пов'язаних з цим значних економічних витрат [3], серед яких витрати на паливо (вугілля, природній газ, нафту або ядерну енергетику).

В рамках досягнення необхідного ефективного економічного рішення для оптимізації процесів в електроенергетичні й системі, що складається із високовольтних ліній, підстанцій, низьковольтних ліній та трансформаторів (рис. 1.1), були розроблені оптимізаційні алгоритми для забезпечення роботи електроенергетичної системи на основі традиційних оптимізаційних методів [4].

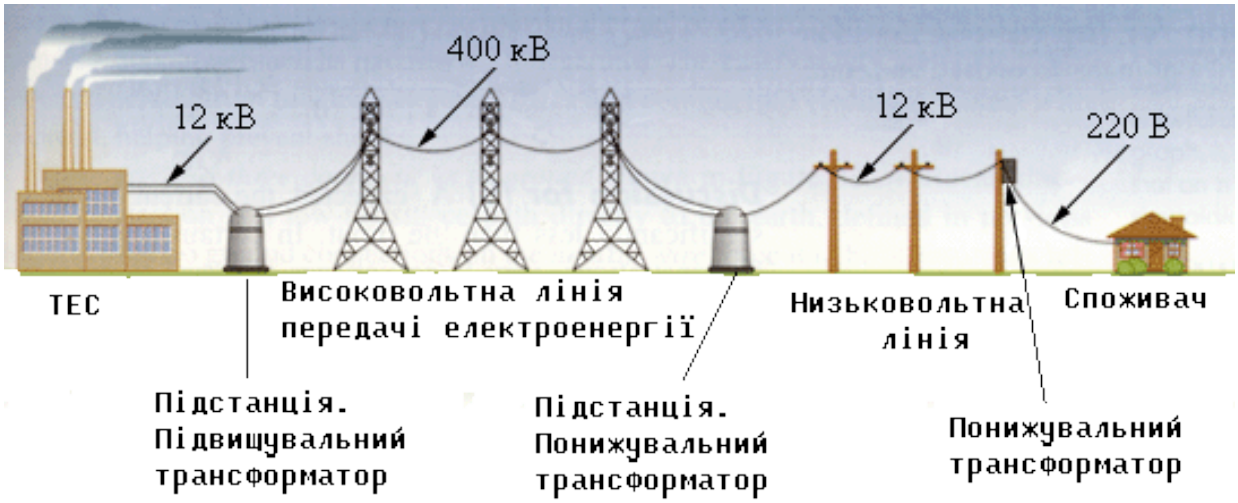

Рисунок 1.1 – Генерація, передача та розповсюдження електроенергії

Ефективне функціонування генераторів в електромережі є результатом кількох задач оптимізації, які взаємодіють між собою [5]. До них відносять задачі вибору складу, що відома в світовій практиці як Unit commitment problem (UCP),

гідротермічного планування – Hydro-thermal scheduling (HTS), економічного розподілу навантаження (ЕРН) – Economic Load Dispatch (ELD) та динамічного розподілу навантаження (ДРН)– Dynamic Load Dispatch (DLD). На даний момент існує не так багато досліджень задачі ДРН відносно інших задач, що зображено на рисунку 1.2. Дані для побудови діаграми базуються на огляді бази наукових робіт із IEL та Science Direct.

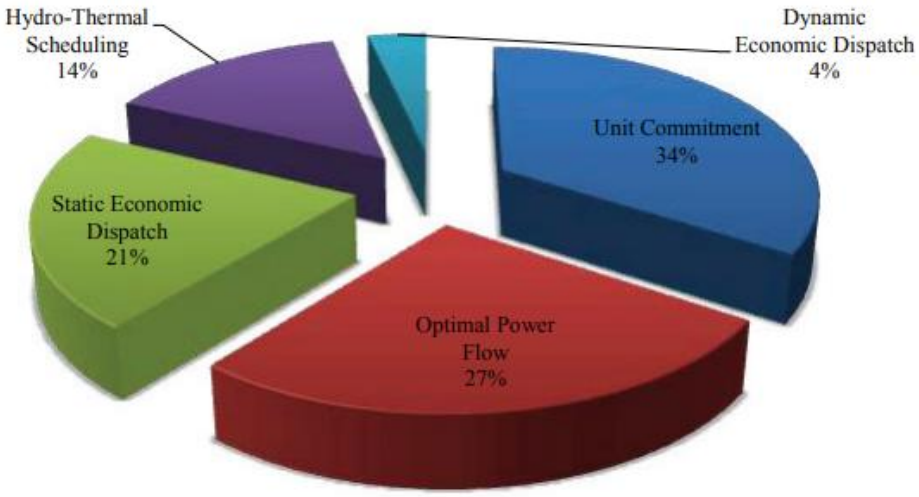

Рисунок 1.2 – Схема розподілу наукових праць

Розв’язок задач ЕРН та ДРН полягає в знаходженні мінімального значення цільової функції, яка включає в себе необхідні параметри функціонування електроенергетичної системи в допустимій області пошуку, яка обмежена вимогами, що накладаються на роботу цієї системи. В загальному вигляді методи класифікують наступним чином: лінійне програмування, нелінійне програмування, стохастичне програмування та мета-евристика. Окремо виділимо категорію методів, що мають назву гібридні технології.

1.1 Методи лінійного програмування

Прості оптимізаційні методи, в яких цільова функція і область пошуку визначаються лінійними залежностями. Методи лінійного програмування застосовувались для різних задач електроенергетики, таких як задачі знаходження економічного розподілу електроенергії, проектування і функціонування

електроенергетичної системи, узгодженості її захисту, забезпечення графіка навантаження, оцінювання стану системи [6-8]. До методу лінійного програмування належить також методи цілочислового програмування, в яких всі або деякі змінні визначені як дискретні цілочислові величини. Цей метод використовується для оцінки безпеки енергетичних систем, для оптимізації та проектування ліній електропередач, аналізу надійності, проектування розподілених систем та менеджменту навантаження [9].

Щоб застосувати метод лінійного програмування до задач економічного розподілу навантаження (ЕРН) та динамічного розподілу навантаження (ДРН), їхні цільові функції, які мають нелінійний характер, необхідно звести до лінійного. Виконавши аналіз робіт [10] можна зробити висновок, що метод показує прийнятні результати для задачі ЕРН, коли в замкненій електроенергетичній системі працює до 10 електростанцій із врахуванням рівняння балансу енергетичної системи та обмеження на максимальну та мінімальну робочу потужність електростанції.

Метод лінійного програмування у [11] застосовують для розв’язування задачі ДРН, не враховуючи обмежень, які пов’язанні із роботою вузлів (клапанів) електростанції. В роботі представлені два підходи до розв’язування задачі. Перший підхід в порівнянні із другим є більш швидкий і менш точний. Якщо ж порівнювати метод лінійного програмування із іншими методами в контексті задач ЕРН та ДРН, метод лінійного програмування, як правило, буде мати низький показник ефективності обчислення [12].

1.2 Методи нелінійного програмування

Оптимізаційні методи, в яких цільова функція задана нелінійною залежністю при допустимій області пошуку, яка може бути виражена як лінійними так і нелінійними залежностями. Однак, вважається, що знаходження оптимального розв’язку нелінійної цільової функції з нелінійними обмеженнями є дуже складною задачею. Тому при розв’язуванні такого типу задач область допустимого пошуку задається лінійною залежністю.

Цей метод інтенсивно використовується для оптимізаційних задач динамічної безпеки електричних систем, контролю перетоків реактивної потужності, проектування і функціонування електричних системи, оптимального потоку потужності в системі, оптимального розміщення засобів компенсації реактивної потужності та оптимальної локації джерел електроенергії [14, 15].

Для розв'язування задачі ЕРН були застосовано ряд підходів, що відносяться до нелінійного програмування, а саме метод невизначених множників Лагранжа (Lagrange multiplier) [16] та метод гілок та меж (Branch and bound) [17]. Але, з огляду на особливості форми області значень цільової функції задач ЕРН та ДРН, зазвичай методи затримуються в локальних оптимумах цільової функції оптимізаційної задачі, що виникає при формалізації проблеми. Крім того, вхідні та вихідні характеристики сучасних генераторів за своєю природою є нелінійними з різних причин, серед яких ефект різнородного палива, нерівномірного навантаження вузлів (клапанів), обмеження швидкості зміни потужності та інші. Ці характеристики і призводять до виникнення цілого ряду локальних оптимумів, які створюють проблеми із знаходженням глобального оптимуму.

1.3 Методи стохастичного програмування

Оптимізаційні методи, які використовуються для розв'язання задач, що містять в собі невизначеність, в яких цільова функція задається імовірнісною функцією. Ці методи також називаються методами динамічного програмування (Dynamic Programming) [18]. Цей метод широко використовується для розв'язання задач оптимізації, але знаходження чисельного розв'язку вимагає значних обчислювальних розрахунків, які збільшують імовірність отримання квазі-оптимальних результатів, які, здебільшого, обумовлені розмірністю змінних. Наприклад, цей метод був використаний для оптимізації потужності джерел генерації електроенергії та в задачі ЕРН [19].

Метод динамічного програмування застосовують для розв'язання задачі ДРН із врахуванням фізичних та експлуатаційних обмежень роботи електростанції [20].

Цей підхід так само як і методи нелінійного програмування часто знаходить локальні оптимуми.

Новий математичний підхід заснований на рекурсивному динамічному програмуванні (Recursive Dynamic Programming) реалізований для задачі ЕРН [21]. Результати моделювання показують, що модифікований метод динамічного програмування може запропонувати більш рентабельні витрати виробництва за незначний час обчислення, ніж ті, що були отримані завдяки генетичному алгоритму.

Покращений метод динамічного програмування для задачі ЕРН із врахування втрат робочої потужності при проходженні в електромережі запропонували у роботі [22]. Метод може застосовуватися до високо масштабних систем до 40 генераторів. Автори статті навели детальне математичне виведення підходу рекурсивного динамічного програмування. Результати випробувань показують, що запропонований підхід може забезпечити більш якісні розв’язки із більшою продуктивністю. Метод порівнювали із генетичним алгоритмом та алгоритмом оптимізації роєм частинок.

1.4 Методи мета евристичної оптимізації

Оптимізаційні методи, в яких оптимальне значення цільової функції при допустимій області пошуку знаходиться методами нечіткої логіки при використанні алгоритмів, які реалізуються на основі систем штучного інтелекту. Головною особливістю цих методів є їх надзвичайна гнучкість при розв’язуванні задач оптимізації, які мають область допустимого пошуку, що виражено різноманітними математичними обмеженнями (як лінійними, так і нелінійними). При цьому оптимальний розв’язок задається множиною простору розв’язків [23]. Існує питання чому мета-евристики стали надзвичайно поширеними.

По-перше, мета-евристики досить прості. Вони, як правило, пов'язані з фізичними явищами, поведінкою тварин або еволюційними концепція. Простота дозволяє вченим з інформаційних технологій моделювати різні природні концепції,

розробляти нові мета-евристики, схрещуючи дві або більше мета-евристики, або поліпшуючи існуючі. Крім того, простота допомагає досить швидко застосовувати їх у вирішенні найрізноманітнішого роду задач.

По-друге, гнучкість даного роду алгоритмів означає їх застосування при вирішенні різного роду проблем без будь-яких особливих змін у власній структурі. Мета-евристики легко застосовуються до вирішення різних задач, оскільки вони, зазвичай, їх у вигляді чорного ящика. Іншими словами, тільки вхід і вихід системи мають важливе значення для мета-евристики. Таким чином, все, що необхідно, це знати, як представити проблему в контексті обраної мета-евристики.

По-третє, більшість мета-евристики не має необхідності у пошуку похідних функцій. На відміну від підходів, що за основу мають градієнтні методи, мета-евристика оптимізує проблеми стохастично. Процес оптимізації починається з випадкового рішення чи набору таких рішень, і немає необхідності обчислювати похідну цільової (чи якоїсь іншої) функції, щоб знайти оптимальне рішення. Це робить мета-евристики зручними у використання при складних (в контексті часу обчислення похідної) функціях або при відсутності інформації про їхній вигляд.

По-чертверте, мета-евристики мають непогані можливості уникнення локальних оптимумів у порівнянні з традиційними методами оптимізації. Це пов'язано з стохастичною природою даних алгоритмів, що дозволяє уникнути застою в локальних оптимуму і здійснювати пошук по всьому простору станів. Вигляд простору станів реальних проблем зазвичай невідомий, має складний вигляд, велику кількість локальних оптимумів та/або плато, через що використання мета-евристик досить непоганий варіант для розв’язання даних задач.

До мета-евристичних методів оптимізації належать: генетичний алгоритм (Genetic Algorithm або GA), алгоритм диференційної еволюції (Differential Evolution або DE), алгоритм оптимізації мурашиної колонії (Ant Colony Optimization або ACO), алгоритм методу рою частинок (Particle Swarm Optimization або PSO) алгоритм вовчої зграї (Grey Wolf Optimizer або GWO), алгоритми оптимізації мурашиними колоніями (Ant Colony Optimization або ACO), штучні нейронні мережі (Artificial neural network або ANN) та інші. Згадані вище мета-евристичні методи

натхненні алгоритмами, які реалізуються для опису функціонування біологічних систем.

Зокрема, генетичний алгоритм та алгоритм диференційної еволюції базуються на уяві про загальні принципи механізмів еволюційної біології. Так, в процесі еволюції живих організмів, нащадки наслідують риси, які є оптимальними при заданих умовах. При цьому генетичний алгоритм призначений для оптимізації функцій дискретних змінних, в ньому акцентується увага на рекомбінаціях геномів, тобто цей алгоритм оперує передачею генетичної інформації біологічної популяції [24]. Алгоритм диференційної еволюції направлений на передачу фенотипної інформації [23].

Генетичний алгоритм застосовують не тільки для розв'язування задачі ЕРН, а і для ДРН із врахуванням щогодинної зміни втрат електроенергії в електромережі [24]. Дослідники запропонували паралельний мікро-генетичний алгоритм [25], що використовує стратегії балансування навантаження та міграції серед процесорів, для задачі ДРН. За допомогою різних стратегій міграції, алгоритм дає на виході кращий розв'язок, ніж звичайний генетичний алгоритм.

При використанні генетичного алгоритму ефективний розв'язок знаходиться швидко, проте цей метод має деякі негативні аспекти, а саме:

– збіжність наближається до локального розв'язку швидше, ніж до глобального. Алгоритм нехтує негативним розв'язком, проходячи лише через хороший. Ця особливість є основним недоліком еволюційних методів;

– складність проходження ряду динамічних даних;

– в окремих оптимізаційних задачах, при достатніх потужностях для обчислень, простіший метод оптимізації дає можливість досягнути кращого результату, ніж той, який реалізується за допомогою генетичного алгоритму [3].

Диференційна еволюція, як і генетичний алгоритм, здатна знайти ефективний розв'язок для задачі ДРН, враховуючи обмеження, що накладаються на фізичні властивості роботи електростанції. Зміна коефіцієнта масштабування на гаусову випадкову величину – покращує ефективність пошуку алгоритму диференційної еволюції [26]. Алгоритм покращують шляхом введення методу евристичного

кросовера та оператора зміни гена [27-29], але ці зміни негативно впливають на час роботи алгоритму.

Еволюційне програмування – це стохастична глобальна технологія пошуку, яка наголошує на поведінці між батьками та їх нащадками, а не намагається емалювати специфічні генетичні оператори, як це робить генетичний алгоритм. Еволюційне програмування починається з популяції випадкових згенерованих кандидатів (батьків) і знаходить рішення паралельно з використанням процесу оцінювання. Початкова популяція цільових векторів (батьків) є рівномірною випадково сформованою в межах допустимого діапазону. Нова популяція розв’язків (нащадків) створюється мутацією, яка модифікує гени батьків за допомогою гаусової випадкової величини. Порівняно із генетичний алгоритмом, алгоритм еволюційного програмування вимагає менше часу для обрахування результатів, завдяки тому що не використовує оператор кросовера.

Даний алгоритм може розв’язувати задачі оптимізації з негладкими цільовими функціями. Його використовували для розв’язування задачі ДРН з гладкою [30-32] та негладкою [33-35] функцією вартості

Алгоритм імітації відпалу – загальний алгоритмічний метод розв'язання задачі глобальної оптимізації, особливо дискретної та комбінаторної оптимізації, в якому процедура пошуку глобального розв'язку імітує фізичний процес відпалу. Для завдань, коли важливіше знайти остаточний глобальний оптимум, ніж знайти точний локальний оптимум за фіксований часу, імітований відпал може бути кращим, ніж альтернативи, такі як градієнтний спуск. Реалізацію алгоритму імітації відпалу для розв’язування задачі ДРН із врахуванням втрат електроенергії та із обмеженнями, що характеризують роботу вузлів, представили у роботі [36]

Нейронні мережі є іще одним популярним методом для розв’язування задач ЕРН та ДРН. Штучні нейронні мережі – це математична модель, а також її програмна та апаратна реалізація, побудована за принципом біологічних нейронних мереж – мереж нервових клітин живого організму. Нейронні мережі можна розділити на кілька типів відносно типів даних, що подаються на вхід мережі, та процедур навчання. Нейронна мережа Хопфілда – це тип рекурентної, повнозв'язної,

штучної нейронної мережі з симетричною матрицею зв’язків [37]. Щоб покращити розв’язок, який отримує нейронна мережа на виході після розв’язання задачі ДРН, було запропоновано використовувати безпечні константи, а додавання ймовірнісного шуму - зменшує ймовірність попадання в локальний оптимум [38]. У роботі [39] застосовано метод повторного відправлення.

Основними перевагами штучних нейронних мереж є його здатність паралельного обчислення та швидке отримання розв’язку за допомогою навченої нейронної мережі. До недоліків відносять довгий час навчання та складний підбір структури нейронної мережі [36].

Ідеї деяких сучасних алгоритмів оптимізації навіяні природою [40]. Серед них виділяються такі, які відносять до ройового інтелекту (Swarm Intelligence) - дисципліни, що має справу з природними та штучними системами, які складаються з багатьох індивідів (елементів, агентів, частинок), координація/узгодження дій між якими здійснюється шляхом децентралізованого управління та самоорганізації. Зокрема, ця дисципліна зосереджується на колективній поведінці, що є результатом локальних взаємодій окремих особин між собою та з навколишнім середовищем. Прикладами таких систем є колонії мурах і термітів, косяки риб, зграї птахів, стада наземних тварин. Серед відомих алгоритмів ройового інтелекту в комбінаторній оптимізації найбільший інтерес становлять алгоритми оптимізації мурашиною колонією (Ant Colony Optimization), оптимізації роєм частинок (Particle Swarm Optimization), бджолині алгоритми (Artificial Bee Colony).

Еволюційні процеси і розгляд поведінкових ситуацій належить до біологічно-імітаційних алгоритмів при знаходженні оптимальних рішень. Серед цих методів для розв’язання задач оптимізації найкращі результати при знаходженні оптимального розв’язку цільової функції досягаються за допомогою *методу оптимізації роєм частинок* (ОРЧ) порівняно з іншими імітаційними алгоритмами [41].

Цей алгоритм інспірований спостереженнями за особливостями соціальної поведінки живих організмів в колективі (косяк риб, зграя птахів). Індивідуальна частинка колективу співвідносить своє оптимальне положення з найкращим

колективним положенням рою. Головною властивістю цього методу є те, що складність і нелінійність задачі не впливають особливим чином на знаходження оптимального значення цільової функції [42].

До переваг алгоритму ОРЧ можна віднести:

- використання пам’яті у алгоритмі є більш ефективним за рахунок того, що кожна частинка здатна запам’ятовувати як своє найкраще попереднє положення, так і найкраще положення свого сусіда;

- алгоритм характеризується ефективною варіативністю.

Це обумовлено тим, що рій частинок використовує найбільш ефективну інформацію для визначення напрямку оптимального розв’язку, як це притаманно поведінці соціальних спільнот.

За останні два десятиліття для методу рою частинками розробили багато модифікацій [43-44]. Для розв’язування задачі ЕРН в статті [45] представлені три вдосконалені алгоритми оптимізації роєм часток.

Оптимізація мурашиними колоніями – мета-евристичний підхід до вирішення важких комбінаторних задач оптимізації. Джерелом натхнення для алгоритму стала поведінка мурах, які використовують феромон як засіб зв’язку. Кожна мураха будує розв’язок для задачі та використовує зібрану інформацію [46] (рис. 1.3). Вони застосовуються для розв'язання багатьох типів оптимізаційних задач, починаючи з класичної задачі комівояжера.

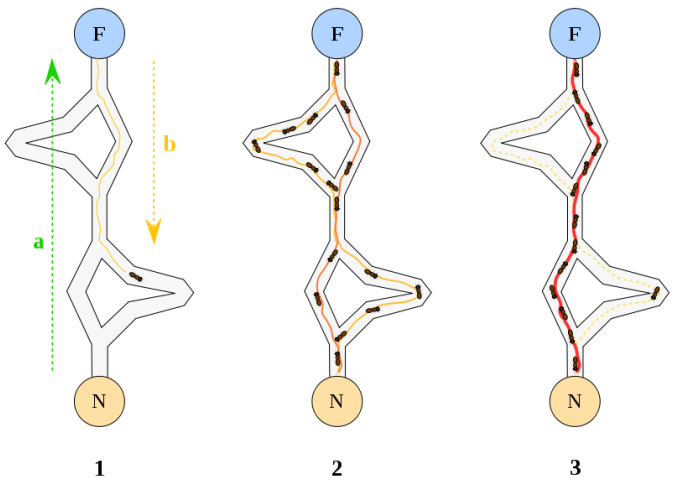

Рисунок 1.3 – Схема пошуку мурахами найкоротшого шляху

Для задачі ЕРН реалізовано алгоритм оптимізації мурашиними колоніями з використанням елітарної стратегії [47]. Ідея елітарної стратегії в контексті «Мурашиної системи» полягає в тому, щоб приділити додатковий акцент на найкращий шлях, виявлений на даному етапі після кожної ітерації. Коли рівень слідів оновлюється, оновлюється і елітарні мурахи. Даний спосіб був застосований для 5 електростанції. Результати порівнювали з класичним генетичним алгоритмом та оптимізацією роєм частинок.

В 2014 році з’явилося нове сімейство алгоритмів, що зв’язані єдиною назвою *оптимізація вовчою зграєю* (Grey Wolf Optimizer, GWO), які ґрунтуються на соціальній ієрархії та мисливській поведінці зграї сірих вовків [48]. Використовуючи термінологію автора, пояснимо принципи організації вовчої зграї. Сірі вовки в основному віддають перевагу проживанні в зграї. Розмір групи складає переважно від 5 до 12 особин. Найцікавішим є те, що вони мають дуже строгу соціальну ієрархію домінуючу, як показано на рисунку 1.4.

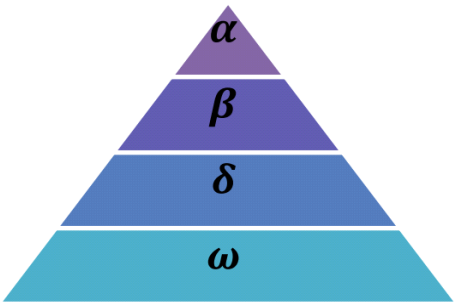

Рисунок 1.4 – Схема системи ієрархії сірих вовків (значущість зменшується зверху вниз)

Вожака зграї називають альфою. Альфа в основному відповідає за прийняття рішень про полювання, вибору місць для ночлігу, часу відпочинку і так далі. Рішення альфи є наказом зграї. Проте, свого роду демократична поведінка також спостерігається, в якій альфа слідує за іншими вовками зграї. Альфу також називають домінуючим вовком, так як його наказ виконується усією зграєю. Цікаво, що альфа не обов'язково найсильніший член зграї, але найкращий у сенсі управління зграєю. Це показує, що організація і дисципліна зграї є набагато важливішою, ніж її сила.

Другу сходинку ієрархії сірих вовків посідає група, яка має назву «бета». «Бета» - підлеглі вовки, які допомагають альфа в процесі прийняття рішень або інших видах діяльності. Бета-вовк може бути чоловічої чи жіночої статі, і він, ймовірно, є найкращим кандидатом, щоб бути альфа в разі, якщо один з альфа-вовків помирає або стає дуже старим. Бета-вовк повинен поважати альфа, але може віддавати накази іншим вовкам нижчого рівня. Він грає роль радника альфа і слідкує за дисципліною у зграї. Бета ретранслює команди альфа зграє і доповідає про стан їх виконання.

Найнижчу сходинку в ієрархії посідає омега. Ці вовки завжди повинні підкорятися всім іншим домінуючим вовкам. Вони є останніми вовками, яким дозволено їсти. Може здатися, що омега не є важливим членами в зграї, але було виявлено, що в разі втрати омеги, зграя стикається з внутрішніми роздором та проблемами. Це пов'язано з відтоком насильства. У деяких випадках омега також є нянями.

Проміжною ланкою між омегою та альфою і бетою є дельта або іще цих вовків називають підлеглими. Дельта вовки повинні підкорятися альфам і бетам, але вони домінують над омегою.

Основними етапами полювання сірих вовків є:

– відстеження, переслідування та наближення до здобичі.

– захоплення, охорона та переслідування здобичі, доки вона не перестане рухатися.

– атака на здобич.

Цей метод застосовують для розв’язування різноманітних задач, таких як наприклад навчання нейронної мережі [49] чи задачі економічного розподілу навантаження [50-51]. У роботі [52] запропоновано модифікацію АВЗ для задачі економічної утилізації викидів. Автор статті зазначив, що метод оптимізації вовчою зграєю страждає від проблеми стагнації (застою) через менш ефективні локальні оптимуми.

1.5 Гібридні технології.

Хоча методи штучного інтелекту здаються перспективними для розв’язування задачі ДРН, однак, зазвичай, отримання достатньо точних розв’язків пов’язане із значними обчислювальними затратами [53]. Гібридні методи інтегрують два або більше оптимізаційні методи, щоб об'єднати їх сильні сторони. Такі гібридні методи виявилися ефективними при пошуку глобального оптимального рішення для задач ЕРН та ДРН з гладкою та негладкою функціями вартості. У гібридних методах спочатку для пошуку використовується один або декілька методів, щоб знайти початкове оптимальне розв’язку, після чого інші методи використовуються для його покращення та отримання остаточного рішення.

У роботі [54] подав гібридний підхід нейронної мережі Гопфілда та квадратичного програмування (HNN-QP) для розв’язування проблеми ДРН із врахуванням втрат при передачі електроенергії. Гібридний алгоритм заснований на застосуванні нейронної мережі Гопфілда як процедури базового пошуку, щоб знайти наближено оптимальний для задачі без врахування обмеженої швидкості зміни робочої потужності. Тоді на задачу знову накладаються дані обмеження і використовується QP для подальшої оптимізації. У роботі [55] запропоновано комбінований генетичний та імітаційний алгоритм відпалу (GA-SA) для вирішення задачі ДРН з немонотонними та монотонно зростаючими функціями додаткових витрат. Алгоритм імітації відпалу використовується для надання базового рішення для генетичного алгоритму, щоб зменшити зусилля пошуку для досягнення оптимального рішення. Даний рішення показало кращі результати, ніж кожний метод окремо.

Гібридні генетичні алгоритми з градієнтним методом успішно використовуються для отримання ефективного рішень для задачі ДРН [56], [57]. У роботі [58] запропоновано гібридний генетичний алгоритм для вирішення проблеми ДРН з транспортними втратами. Запропонована гібридна схема розроблена таким чином, що генетичний алгоритм виконує функцію пошуку базового рішення, а потім застосовуються градієнтний метод для точного настроювання. У роботі [59]

запропонували розслаблений гібридний алгоритм та градієнтну техніку для вирішення проблеми ДРН з транспортними витратами. Запропонована гібридна схема побудована таким чином, що генетичний алгоритм виконує пошук базового рішення, швидко приймає рішення про напрямок градієнта, щоб швидко піднятися на потенційний «пагорб».

У роботі [60] автор реалізував гібридний EP-SQP для розв'язування задачі ДРН із врахуванням особливості роботи вузлів (клапанів) при обрахуванні цільової функції. Гібридний метод EP-SQP включає в себе простий алгоритм еволюційного програмування як головного оптимізатора та метод послідовного квадратичного програмування як локальний оптимізатор для точного налаштування пошуку першого алгоритму. Для ілюстрації ефективності запропонованого методу було використано 10-елементну систему, порівняння проводилися лише з результатами, одержаними із використанням еволюційного алгоритму та з методом послідовного квадратичного програмування.

У роботі [61] автор використав гібридний підхід, інтегруючи оптимізацію роєм частинок із методом послідовного квадратичного програмування (PSO-SQP) для вирішення проблеми ДРН із врахуванням особливості роботи вузлів (клапанів) при обрахуванні цільової функції та транспортними втратами. У запропонованому алгоритмі оптимізації роєм частинок застосовується як процедура пошуку базового рівня, а метод послідовного квадратичного програмування використовується для точного налаштування першого алгоритму.

У роботі [62] автори додали нові обмеження до формулювання задачі, представлені в [63], ввівши прямий резерв та обмеження на певні діапазони робочих потужностей. Задача ДРН також розв'язується гібридною технікою PSO-SQP. У роботі [64] представлений модифікований гібрид EP-SQP для розв'язування тієї ж задачі, представленої в [62]. Запропонований алгоритм реалізовано так, що агенти еволюційного програмування вільно досліджують розв'язки, після чого метод послідовного квадратичного програмування буде використано для оптимізації агентів. Ефективність запропонованих методів, представлених у [62], [63] та [64],

продемонстрована на системі із десятьма генераторами та порівнюється з іншими методами.

Реалізований алгоритм у [65] пов'язаний із гібридним покращенням диференціальної еволюції (IDE) та r-алгоритмом Шора для розв’язування задачі ДРН із врахуванням особливості роботи вузлів (клапанів). IDE застосовується як пошуку базового рівня, який надає область біля глобального оптимуму, а r-алгоритм Шора використовується як локального пошуку для точного налаштування ефективного рішення.

Автор у роботі [66] реалізував гібридний генетичний алгоритм з квазі-симплексною технікою для розв’язування задачі ДРН із невизначеностями в коефіцієнті функції витрат. Невизначеності представляли нечіткі числа. Чжан [67] запропонував гібридний генетичний алгоритм із натуральними числами у коді з квазі-симплексною технікою для вирішення ДРН із врахуванням особливості роботи вузлів (клапанів). У [66] та [67] запропоновані алгоритми генерують нащадків за допомогою генетичного алгоритму та квазі-симплексної техніки паралельно.

У роботі [68] автори навели гібридну техніку інтегруючи еволюційне програмування та оптимізацію роєм частинок з методом послідовного квадратичного програмування для задачі ДРН із транспортними втратами, з врахуванням особливості роботи вузлів (клапанів), з обмеженням швидкості зміни робочої потужності та з обмеження на певні діапазони робочих потужностей. Як еволюційне програмування, так і оптимізація роєм частинок використовуються як основний оптимізатор для пошуку майже глобального рішення, тоді як метод послідовного квадратичного програмування використовується для точного налаштування.

При застосовуванні алгоритму оптимізації мурашиними колоніями до задачі ДРН виникає ряд недоліків пов’язаних із стратегією та повільним сходженням до ефектного розв’язку. Таким чином був розроблений новий алгоритм із додаванням методу диференційної еволюції (DEACO). Порівняльні дослідження між DEACO та звичайним алгоритмом оптимізації мурашиними колоніями показали, що новий алгоритм подолав слабкі сторони класичного алгоритму [69].

Висновки до розділу

У сфері енергетики прагнуть досягти ефективної експлуатації електромереж, беручи до уваги проблеми збільшення вартості палива та попиту на електроенергію. Оскільки задачі ЕРН та ДРН мають широке застосування на практиці, вони є важливими та актуальними у галузі енергетики.

Для проведення аналізу наукових робіт було прийнято рішення розбити методи розв’язування задачі розподілу навантаження на такі категорії: лінійне програмування, нелінійне програмування, стохастичне програмування та мета-евристика та гібридні технології.

У розділі наведено переваги та недоліки кожної категорії. З огляду на особливості форми області значень цільових функцій оптимізаційних задач, зазвичай метод лінійного, нелійного та стохастичного програмування затримуються в локальних оптимумах цільової функції. Крім того, вхідні та вихідні характеристики сучасних генераторів за своєю природою є нелінійними з різних причин, серед яких ефект різнорідного палива, нерівномірного навантаження вузлів (клапанів), обмеження швидкості зміни потужності та інші. Ці характеристики і призводять до виникнення цілого ряду локальних оптимумів, які створюють проблеми із знаходженням глобального оптимуму.

Серед представників метаевристичних методів можна виділити АВЗ та ОРЧ для задачі ЕРН. Широкої популярності за остання десятиліття набула категорія гібридних методів. На практиці вони демонструють кращі результати в порівняння із класичними методами.

2 ПОСТАНОВКА ТА ФОРМАЛІЗАЦІЯ ЗАДАЧІ РОЗПОДІЛУ НАВАНТАЖЕННЯ

2.1 Змістова постановка задачі

Розглянемо замкнену систему, що складається із декількох генераторів, кількість яких позначимо N генераторів, що зображена на рисунку 2.1. Кожна із них працює впродовж 24 годин, генеруючи потужність P_i (МВт) $i = \overline{1, N}$. Розглядаються системи, у яких застосовують різні типи паливних ресурсів (гідроенергетика, газ, пар, дизельне паливо, ядерне паливо, вугілля і т.д.), що необхідні для роботи генераторів.

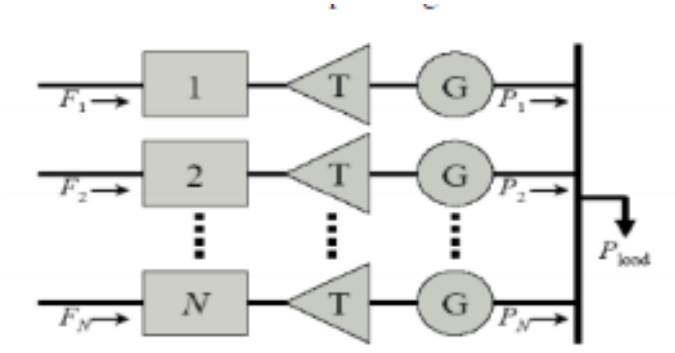

Рисунок 2.1 – Замкнена система із N генераторів

З огляду на особливості внутрішньої будови генератора та взаємодію із електромережею, накладаються умови на генерацію робочої потужності P_i, а саме встановлюється мінімальна ($P_i^{\min}$) та максимальна ($P_i^{\max}$) межі робочих потужностей. Також генератор не може продукувати електроенергію з потужністю в певних проміжках (заборонених зонах).

В залежності від часу доби змінюється потреба в потужності електромережі P_D, з огляду на зміну діяльності населення. На кількість згенерованої робочої потужності P_i впливають загальні втрати P_L. До них входять постійні втрати, що описуються коефіцієнтом B_{00}, та динамічні втрати, що залежать від P_i.

Загальна вартість електроенергії включає вартість праці, вартість палива, технічну підтримку генераторів та втрати електроенергії. Як правило, витрати на

оплату праці, постачання палива та технічна підтримка становлять фіксований відсоток від вартості.

Задача полягає у пошуку розподілу навантажень між генераторами, що задовольняє всім вище зазначеним умовам та надає найменшу вартість електроенергії.

Головна мета енергоекономічного розподілу полягає у знаходженні загальної потужності, що виробляють генератори, та мінімізувати експлуатаційні втрати. Окрім основної мети також є ряд завдань:

– мінімізація викидів, а саме газоподібних викидів SO2, NOx, CO та CO2,що потрапляють в атмосферу завдяки тепловим електростанціям;

– максимізація прибутку шляхом оптимізації виробництва;

– підтримування стійкості системи;

– дотримання експлуатаційних обмежень, що накладаються на роботу генераторів.

2.2 Математична модель задачі

2.2.1 Задача економічного розподілу навантаження

Ціль класичної задачі економічного розподілу навантаження полягає у знаходженні мінімального значення робочої потужності для кожного генератора [19]:

$$f = \sum_{i=1}^{N} F_i(P_i) \to \min, \tag{2.1}$$

де, $F_i(P_i)$ – вартість палива, яке використовують для генерації робочої потужності P_i , N – кількість генераторів. Як правило, експлуатаційні втрати кожного генератора при генерації конкретної вихідної потужності моделюються як:

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2, \tag{2.2}$$

де a_i, b_i, c_i – коефіцієнти витрат i-о генератора, P_i – це робоча потужність i-о генератора в МВт.

Задача ЕРН підпорядкована ряду обмежень. Ці обмеження поділяють на дві категорії:

Вимоги до генераторів.

Рівняння балансу

$$\sum_{i=1}^{N} P_i - P_D - P_L = 0,$$

де P_D – загальна попит на електроенергію; P_L – втрати робочої потужності. P_L розраховується на основі матриці коефіцієнтів втрат B та коефіцієнта втрат B_{00}:

$$P_L = \sum_{i=1}^{N}\sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{n} B_{i0} P_i + B_{00}.$$

Вимоги до активного резерву

$$\sum_{i=1}^{N} S_i^t \geq SR^t,\ t = \overline{1,T},$$

де S_i^t – це активний резерв від i-о генератора в момент часу t; SR^T – це активний резерв, що вимагає система від всіх генераторів в момент часу t.

Температурні обмеження генераторів

– мінімальний час роботи – мінімальний час, який має пропрацювати генератор перед вимкненням;

– мінімальний час простою – мінімальний час простою генератора перед повторним увімкненням.

Обмеження на постійну напругу в замкненій системі

У деяких випадках генератори продовжувати свою роботу через вимогу підтримки напруги в середині замкненої системи. Дане обмеження також поширюється на виробництво водяного пару або використання пари генератором.

Обмеження на паливо

Специфіка роботи генераторів встановлює наступні обмеження на паливо:

- паливо має горіти у встановлений час;
- тип палива.

Обмеження генерації робочої потужності

Дане обмеження зумовлено експлуатаційною особливістю генератора та впливає на генеруючи потужність під час переходу від одного виробничого рівня до іншого, при цьому дотримуючись безпечних температурних режимів:

$$P_i^{\min} \le P_i \le P_i^{\max},$$

де $P_i^{\min}, P_i^{\max}$ – це мінімальна та максимальна межі реальних робочих потужностей для i-о генератора.

Обмеження швидкості зміни робочої потужності

Робочі діапазони вузлів генератора обмежені у швидкості. На рисунку 2.2 зображено три можливі варіанти роботи генератора від часу $t-1$ до t, та який перебуває у :

1) постійному робочому стані;
2) зростаючому робочому стані;
3) спадаючому робочому стані.

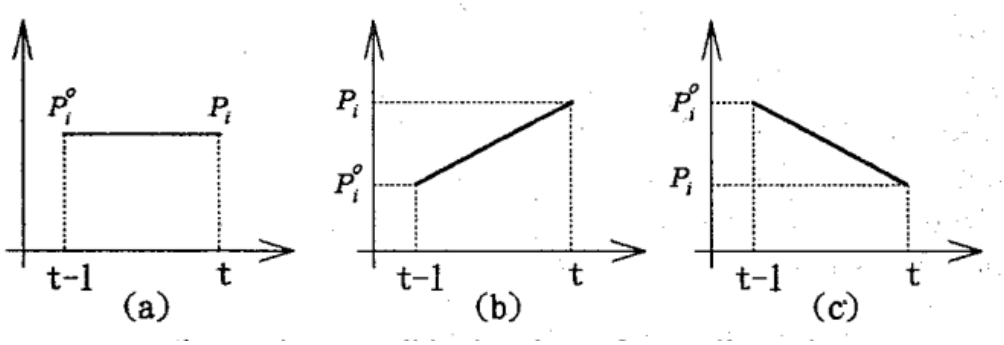

Рисунок 2.2 – Схема можливих комбінацій зміни швидкості робочої потужності від часу роботи

Якщо потужність генератора зростає:

$$P_i - P_i^0 \le UR_i.$$

де UR_i – це верхня межа різниці між потужностями в попередній та поточний періоди. Якщо потужність генератора спадає:

$$P_i^0 - P_i \le DR_i.$$

де DR_i – це нижня межа різниці між потужностями в попередній та поточний періоди.

Заборонені діапазони значень робочої потужності

Через специфіку роботи вузлів генератора та вібрації в підшипниках розглядають заборонені діапазони значень робочої потужності генератора. Практично найкраща економія досягається шляхом уникнення експлуатації в таких місцях протягом всієї операції. Робочі зони (рис. 2.3) i-о генератора описані нижче:

$$\begin{aligned} &P_i^{\min} \le P_i \le P_{i,1}^u, \\ &P_{i,j-1}^l \le P_i \le P_{i,j}^u, \\ &P_{i,k}^l \le P_i \le P_i^{\max}, \\ &j = \overline{2,k}, \end{aligned}$$

де l, u – нижня та верхня межі для j-ї робочої зони, а k – кількість робочих зон.

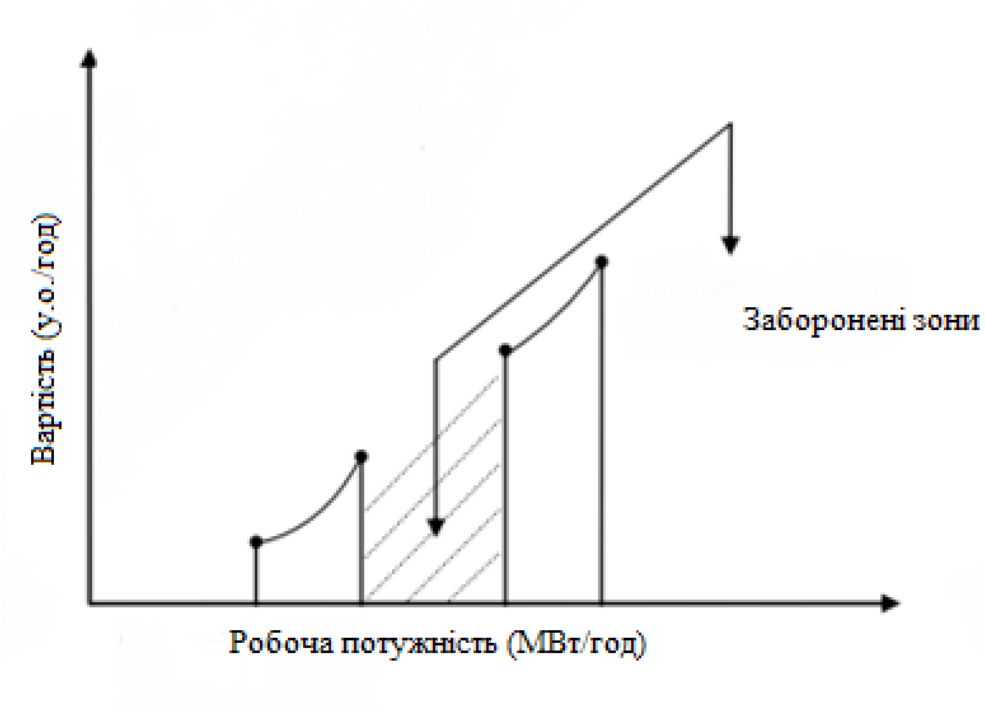

Рисунок 2.3 – Схема заборонених зон

Вимоги до електромережі.

Мережева безпека

Встановлюється ліміт +/- 10% на напругу в середині електромережі, якщо напруга в середині системі в межах від 132 кВ до 220 кВ.

Лінійна потужність під різними потоками та навантаженнями

$$F_l^t \le F_l^{\max},\ l = \overline{1,L},$$

де F_l^t – це активний потік потужності через лінію електропередачі l протягом інтервалу часу t; $F_l^{\max}$ – це верхня межа потоку активної потужності вздовж ліній l, L - кількість ліній електропередач.

2.2.2 Задача економічного розподілу навантаження без врахування втрат

Задачі економічного розподілу навантаження без врахування втрат є спрощеним варіантом задачі ЕРН. Побудуємо математичну модель для даної задачі враховуючи, що функція вартості палива $F(P_i)$ відома для всіх генераторів. Цільова

функція аналогічна до цільової функції задачі ЕКР формула 2.1 та 2.2. Наведемо обмеження, що описують роботу генераторів у замкненій системі.

Рівняння балансу

За умовою задачі необхідно нехтувати втратами робочої потужності, тому сума потужностей всіх генераторів у системі має бути рівною загальному попиту P_D.

$$\sum_{i=1}^{n} F(P_i) = P_D,$$

Обмеження генерації робочої потужності

$$P_i^{\min} \le P_i \le P_i^{\max}.$$

2.2.3 Задача економічного розподілу навантаження із точковим навантаженням клапанів

Функцію вартості палива $F_i(P_i)$ i-о генератора для задачі ЕРН найчастіше апроксимують у вигляді квадратичної або кусково-квадратичної функцій. Про те реальні характеристики функції вартості палива демонструють нелінійний характер та розриви, що пов’язано із фізичним навантаженням на клапали під час спалювання палива.

Загальноприйнята функція вартості палива в електроенергетиці з точки зору виробленої робочої потужності P із врахування ефекту навантаження клапанів для i-о генератора, має наступний вигляд:

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c + \left| e_i \sin(y_i \cdot (P_i^{\min} - P_i)) \right|,$$

де a_i, b_i, c_i – коефіцієнти витрат i-ї електростанції, а e_i та y_i – константи, що показують вплив клапана i-ї електростанції. На рисунку 2.4 зображено функцію вартості палива із (лінія) та без (пунктир) врахування точкових навантажень на клапани .

Рисунок 2.4 – Функція вартості палива

2.2.4 Задача динамічного розподілу навантаження

На відмінну від задачі ЕРН, у задачі ДРН враховує зміну попиту на електроенергію протягом дня [32]. Таким чином, задача з цільовою функцією f, яка характеризуватиме вартість палива в у.о., виглядатиме наступним чином:

$$f = \sum_{t=i}^{T} \sum_{i=1}^{N} F_{ti}(P_{ti}) \to \min, \tag{2.3}$$

де T – кількість запланованих періодів, N – кількість електростанцій, а $F_{ti}(P_{ti})$ – вартість палива, яке використовують для генерації робочої потужності P_i в момент часу t. Беручи до уваги особливості роботи клапанів генераторів, функцію витрат палива i-о генератора у (2.3) можна подати як суму квадратичної та синусоїдальної функції у такій формі:

$$F_{ti}(P_{ti}) = a_i P_{ti}^2 + b_i P_{ti} + c + \left| e_i \sin(y_i \cdot (P_i^{\min} - P_{ti})) \right|,$$

де a_i, b_i, c_i – коефіцієнти витрат i-о генератора, а e_i та y_i – константи, що показують вплив клапана i-о генератора, P_i – це робоча потужність i-о генератора в МВт.

Наведемо обмеження, що накладаються на замкнену систему електромережі.

Рівняння балансу

$$\sum_{i=1}^{N} P_{ti} - P_{tD} - P_{tL} = 0,$$

де P_{tD} – загальна попит на електроенергію в момент часу t; P_{tL} – втрати робочої потужності в момент часу t. Величина P_{tL} розраховується наступним чином:

$$P_{tL} = \sum_{i=1}^{N}\sum_{j=1}^{N} P_{ti} B_{ij} P_{tj} + \sum_{i=1}^{n} B_{i0} P_{ti} + B_{00},$$

де B_{ij} – i, j-й елемент квадратичної матриці коефіцієнтів втрат, що має розмірність $N \times N$, B_{i0} – i-й елемент вектора коефіцієнтів втрат, що має розмірність $1 \times N$, B_{00} – константа втрат.

Обмеження генерації робочої потужності

$$P_i^{\min} \le P_{ti} \le P_i^{\max}, \qquad (2.4)$$

де $P_i^{\min}, P_i^{\max}$ – це мінімальна та максимальна межі реальних робочих потужностей для i-о генератора.

Заборонені зони

Через дію клапанів та вібрації в підшипниках розглядають діапазони значень робочої потужності, при яких робота генератора неможлива або завдає великих збитків. Ці діапазони називають забороненими зонами експлуатації. Практично найкраща економія досягається шляхом уникнення експлуатації в таких місцях протягом всієї операції. Робочі зони i-о генераторaописуються так:

$$P_i^{\min} \le P_{ti} \le P_{i,1}^u,$$
$$P_{i,j-1}^l \le P_{ti} \le P_{i,j,}^u$$
$$P_{i,k}^l \le P_{ti} \le P_i^{\max},$$
$$j = \overline{2,k},$$

де l,u – нижня та верхня межі для j-ї робочої зони, а k – кількість робочих зон.

Обмеження швидкості зміни робочої потужності

$$P_{ti} - P_{(t-1)i} \le UR_i,$$
$$P_{(t-1)i} - P_{ti} \le DR_i, \qquad (2.5)$$

де UR_i та DR_i – це верхня та нижня межа різниці між потужностями в попередній та поточний періоди. Використовуючи обмеження генерації робочої потужності (2.4), обмеження (2.5) можна переписати наступним чином [15]:

$$\max(P_i^{\min}, P_{(t-1)i} - DR_i) \le P_{ti} \le \min(P_i^{\max}, P_{(t-1),i} + UR_i).$$

2.2.5 Зведення задачі ДРН до задачі із неперервним простором визначення

З огляду на те, що обмеження задачі ДРН роблять її простір визначення кусково-неперервним, великий спектр методів (зокрема і методи ройового інтелекту) не можуть бути застосовані оскільки в своїй роботі використовують неперервні функції вибору компонентів розв'язку. Ключовим фактором при застосуванні методів, що базуються на алгоритмах ройового інтелекту, для вирішення задачі ДРН є те, яким чином обходять обмеження задачі.

Більшість задач оптимізації мають обмеження. Простір пошуку для таких задач складається з точок двох типів: можливих та неможливих. Можливі точки задовольняють усі обмеження задачі, а неможливі – порушують хоча б одне з них. Тому рішення або множина рішень, отриманих у якості розв'язку задачі оптимізаційним методом, повинні обов'язково належати до множини можливих точок – задовольняти усі обмеження. У таких випадках часто використовують методи, що базуються на використання штрафної функції. Задача з обмеженнями

може бути перетворена у задачу без них накладаючи штрафи за порушення обмежень, таким чином збільшуючи значення цільової функції. За рахунок цього стає можливим використання наведених алгоритмів.

За останні кілька десятиліть було запропоновано декілька методів для задоволення обмежень у задачах оптимізації [41]. Ці методи можуть бути об'єднані у чотири категорії: методи, які зберігають можливість допустимих розв'язків, штрафні методи, методи, які чітко розрізняють допустимі та недопустимі розв'язки, та гібридні методи.

Коли алгоритми оптимізації використовуються для задач оптимізації із обмеженнями, загальним є вирішення обмежень за допомогою концепцій штрафних функцій (які збільшують значення цільової функції недопустимих розв'язків). Так цільову функцію задачі ДРН у просторі пошуку можна представити у вигляді:

$$f = \begin{cases} f(P_i), \textit{якщо } P_i \in F \\ f(P_i) + penalty(P_i) \end{cases},$$

де $penalty(P_i)$ – позитивно визначена функція для усіх недопустимих P_i. Штрафна функція зазвичай базується на відстані від неможливого розв'язку до найближчого допустимого значення або на зусиллях, необхідних прикласти для переведення недопустимих розв'язку до можливого.

Варто зазначити, що обмеження на операційні потужності генераторів не потребують введення додаткової штрафної функції. Для усунення даного обмеження достатньо виконати приведення усіх недопустимих розв'язків, що порушують верхній операційний ліміт застосувати (2.6), а до тих, що порушують нижній – (2.7).

$$P_i(t+1) = P_i(t) + \beta \cdot rand(0,1), \tag{2.6}$$

$$P_i(t+1) = P_i(t) - \beta \cdot rand(0,1), \tag{2.7}$$

де $\beta \in (0,1)$ – емпіричний коефіцієнт, t – номер поточної ітерації i $rand(0,1)$ – рівномірно розподілена випадкова величина на проміжку від 0 до 1.

Таким чином при отриманні компоненти розв’язку задачі, що не задовольняє даному обмеженню, воно завжди буде зведено до можливого.

У якості фітнес-функції, яка буде використана замість цільової функції (2.3) при визначенні якості отриманого агентом розв’язку, використано таку:

$$f = \sum_{t=i}^{T}\sum_{i=1}^{N} F_{ti}(P_{ti}) + q_1 P_1 + q_2 P_2 + q_3 P_3 \to \min, \qquad (2.8)$$

де q_1, q_2, q_3 – дані коефіцієнти підбираються емпіричним шляхом і в даній роботі використані зі значеннями 50, 1 та 1.

Функція штрафу для рівняння балансу визначається наступним чином:

$$P_1 = (\sum_{t=1}^{T}\sum_{i=1}^{N} P_{ti} - P_{tL} - P_{tD})^2 .$$

Визначимо функцію штрафу для обмеження заборонених зон:

$$P_2 = \sum_{t=1}^{T}\sum_{j=1}^{N} V_{k,j} ,$$

де $V_{k,j} = \begin{cases} 1, \textit{якщо} P_{k,j} \textit{знаходиться в забороненій зоні} \\ 0 \textit{в іншому разі} \end{cases}$.

Функція штрафу для обмеження швидкості зміни робочої потужності:

$$P_3 = \sum_{t=1}^{T}\sum_{j=1}^{N} Q_{k,j} ,$$

де $Q_{k,j} = \begin{cases} 1, \textit{якщо } \Delta P \textit{ порушує задані обмеження} \\ 0 \textit{ в іншому разі} \end{cases}$.

Висновки до розділу

Сформульовано змістовну постановку задачі розподілу навантаження у замкнених енегростистемах. Наведена математична модель для класичної ЕРН та ДРН, яка включає наступні обмеження:

- рівняння балансу;
- вимоги до активного резерву;
- температурні обмеження генераторів;
- обмеження на постійну напругу в замкненій системі;
- обмеження на паливо;
- обмеження генерації робочої потужності;
- обмеження швидкості зміни робочої потужності;
- заборонені діапазони значень робочої потужності;
- мережева безпека;
- лінійна потужність під різними потоками та навантаженнями.

Розглянуто особливості розрахунку функції палива із врахування навантаження на клапани генератора. Описано перехід від задачі із обмеженнями, що характеризують фізичні та експлуатаційні характеристики електростанцій, до задачі без обмежень із використанням функцій штрафів.

3 АЛГОРИТМИ РОЗВ'ЯЗУВАННЯ ЗАДАЧІ

Для розв'язування задач ЕРН та ДРН пропонується розробити алгоритми вовчої зграї, оптимізації роєм частинок та генетичний. Ці популяційні алгоритми об'єднує те, що кінцевим розв'язком є найкращий представник серед сформованих популяцій, зазвичай, серед фінальної.

Розв'язком задачі ЕРН є вектор $X = (X_1, X_{2,}, ..., X_i, ..., X_N)$, де i-а компонента вектора відповідає згенерованому значенню робочої потужності для i-о генератора, де $i = \overline{1,N}$, а N - кількість генераторів. Розв'язком задачі ДРН є вектор $X = (X_{1,1}, X_{1,2}, ..., X_{1,24}, ..., X_{i,1}, ..., X_{i,j}, .., X_{i,24}, ..., X_{N,1}, ..., X_{N,24})$, де i-а компонента вектора відповідає згенерованій робочої потужності для i-о генератора в момент часу $j = \overline{1,24}$.

3.1 Оптимізація роєм частинок

У алгоритмі ОРЧ координати кожної частинки являються собою можливий розв'язок, який пов'язаний із розташуванням та вектором швидкості [41]. На швидкість частинки впливають три компоненти: інерційна, когнітивна та соціальна (рис. 3.1). Інерційний компонент імітує інертну поведінку фізичних об'єктів, що продовжують рухатися деякий час по інерції в попередньому напрямку після рішення змінити його. Когнітивний компонент моделює пам'ять частинки щодо свого попереднього найкращого положення, а соціальна складова – вплив найкращого розв'язку, отриманого набором частинок за весь процес роботи.

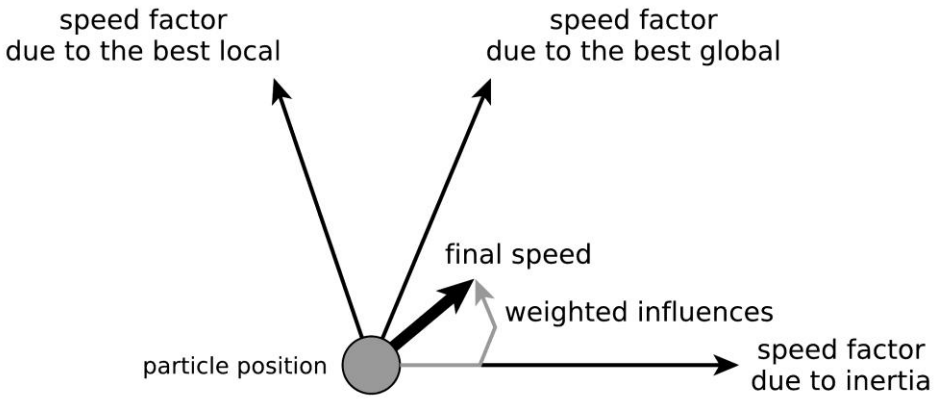

Рисунок 3.1 – Схема впливу компонент на швидкість частинки

Частинки рухаються у багатовимірному просторі пошуку до тих пір, поки вони не знайдуть оптимальний чи близький до нього розв'язок. Модифіковану швидкість кожної частинки можна розрахувати, використовуючи поточну швидкість та відстань від *Pbest* (найкращого розв'язку конкретної частинки) та *Gbest* (найкращого розв'язку рою):

$$V_{ij}^{k} = w \cdot V_{ij}^{k-1} + C_1 \cdot r_1 \cdot (Pbest_{ij}^{k-1} - X_{ij}^{k-1}) + C_2 \cdot r_2 \cdot (Pbest_{ij}^{k-1} - X_{ij}^{k-1}), \qquad (3.1)$$

$$i = \overline{1, N_D},$$

$$j = \overline{1, N_{par}},$$

де k – поточна ітерація, V_{ij}^{k} – швидкість i-ї компоненти j-ї частинки для поточної ітерації k, X_{ij}^{k} – позиція i-ї компоненти j-ї частинки для поточної ітерації k, w – вага інерції, C_1, C_2 – коефіцієнти прискорення, $Pbest_{ij}^{k-1}$ – найкраща позиція i-ї компоненти j-ї частинки до ітерації k, $Gbest_{i}^{k-1}$ – i-а компонента найкращого розв'язку популяції до ітерації k, N_D – кількість компонент (задача ЕРН – кількість генераторів в системі, задача ДРН – кількість генераторів в системі на 24 години), N_{par} – кількість частинок в популяції, r_1, r_2 – випадкові величини рівномірно розподілені на проміжку $[0,1]$. На рисунку 3.2 зображено рух i-ї частинки для k-ї ітерації.

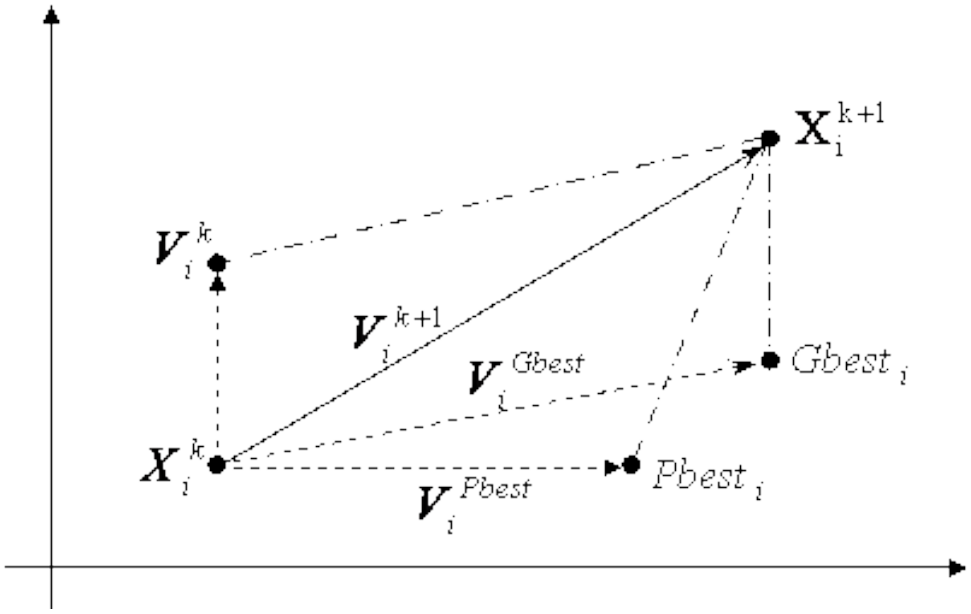

Рисунок 3.2 – Схема руху i-ї частинки для k-ї ітерації

Отримана швидкість із виразу використовується для обчислення позиції частинки:

$$X_{ij}^{k} = X_{ij}^{k-1} + V_{ij}^{k}, \qquad (3.2)$$

Наведемо схему роботи алгоритму оптимізація роєм частинок для задачі розподілу навантаження:

Крок 1. Генерація популяції. Для популяції розміру N_{par} компоненти частинок (робочі потужності) генеруються випадковим чином, їх значення рівномірно розподілені на проміжку від 0 до 1 між мінімальними $P_i^{\min}$ та максимальними $P_i^{\max}$ межами реальних робочих потужностей для i-о генератора. Для N генераторів i-а частинка X_i виглядатиме для задачі ЕРН $X_i = (X_{i1}, X_{i2}, ..., X_{iN})$ та для задачі ДРН $X_i = \left(X_{i1}, X_{i2}, ..., X_{iN^*}\right)$, де $N^* = 24N$, при умові, що генератор працює 24 години, де $X_{ij} = X_j^{\min} + r(X_j^{\max} - X_j^{\min}),\ r \in [0;1]$.

Крок 2. Встановити результат для найкращого розв'язку популяції $Gbest = \infty$ та найкращого розв'язку i-ї частинки $Pbest_i = \infty$, де $i = \overline{1, N}$.

Крок 3. Для кожної частинки X_j, $j = (\overline{1, N_{par}})$ із популяції.

3.1 Визначити значення фітнес-функції за допомогою формули 2.8.

3.2 Якщо нове значення фітнес-функції краще за найкращий розв'язок i-ї частинки $Pbest_i$ до ітерації k, то запам'ятати нове значення в $Pbest_i$.

3.3 Запам'ятати позицію частинки.

Крок 4. Встановити значення для найкращого розв'язку популяції як $Gbest = \min\{Gbest, \min\limits_{i=(1,N_{par})} Pbest_i\}$.

Крок 5. Якщо досягнута максимальна кількість ітерацій, то перейти до кроку 11.

Крок 6. Обрахувати значення швидкості кожної частинки,використовуючи формулу 3.1.

Крок 7. Якщо поточна швидкість частинки більша за максимальну швидкість для j компоненти $V_{ij} > V_j^{\max}$, то $V_{ij} = V_j^{\max}$, де $V_j^{\max} = 1{,}5 \cdot V_{ij}^{k-1}$.

Крок 8.Якщо поточна швидкість частинки менша за мінімальну швидкість для j компоненти $V_{ij} < V_j^{\min}$, то $V_{ij} = V_j^{\min}$, де $V_j^{\min} = 0{,}5 \cdot V_{ij}^{k-1}$.

Крок 9.Змінити позицію кожної частинки, використовуючи формулу 3.2, при цьому якщо робоча потужність X_{ij}^{t+1} виходить за межі реальних робочих потужностей – встановити значення, що наближене до робочої потужності X_{ij}^{t+1}, що задовольняє критеріям.

Крок 10. Перейти на крок 3.

Крок 11. Повернути значення $Gbest$ як найкращий знайдений результат.

3.2 Оптимізація вовчою зграєю

Оптимізацію вовчою зграєю можна описати наступним схему [48].

1. Соціальна ієрархія.

У математичній моделі соціальної ієрархії сірих вовків альфою вважають найкращий розв'язок задачі. Відповідно, другий найкращий розв'язок називають бетою, а третій – дельтою. Розв'язки, які залишилися, приймають як омега. У алгоритмі вовчої зграї оптимізація – це моделювання процесу полювання, яким

займаються альфа, бета та дельта. Омега-вовки лише допомагають у пошуку здобичі.

2. Оточення здобичі

Сірі вовки оточують здобич під час полювання. Поведінка навколишнього середовища моделюється наступним чином:

$$\overline{X}(t+1) = \overline{X_p}(t) - \overline{A}\left|\overline{C}\,\overline{X_p}(t) - \overline{X}(t)\right|,$$

$$\overline{A} = 2\overline{a}\overline{r_1} - \overline{a},$$

$$\overline{C} = 2\overline{r}_2,$$

де $\overline{X}$ – вектор координат розв'язку (омега вовк) на поточній ітерації t, $\overline{X_p}$ – вектор координат одного із найкращих на поточній ітерації розв'язків $p \in \{\alpha, \beta, \delta\}$, $\overline{A}$ – вектор соціальних коефіцієнтів, $\overline{C}$ – вектор когнітивних коефіцієнтів, $\overline{r_1}$ та $\overline{r_2}$ – вектори випадкових величин рівномірно розподілений на проміжку [0,1], $\overline{a}$ – вектор, компоненти якого лінійно поступово зменшуються від 2 до 0 протягом виконання алгоритму.

На рисунку 3.3 зображено етап оточення здобичі вовком та його можливі позиції на наступній ітерації.

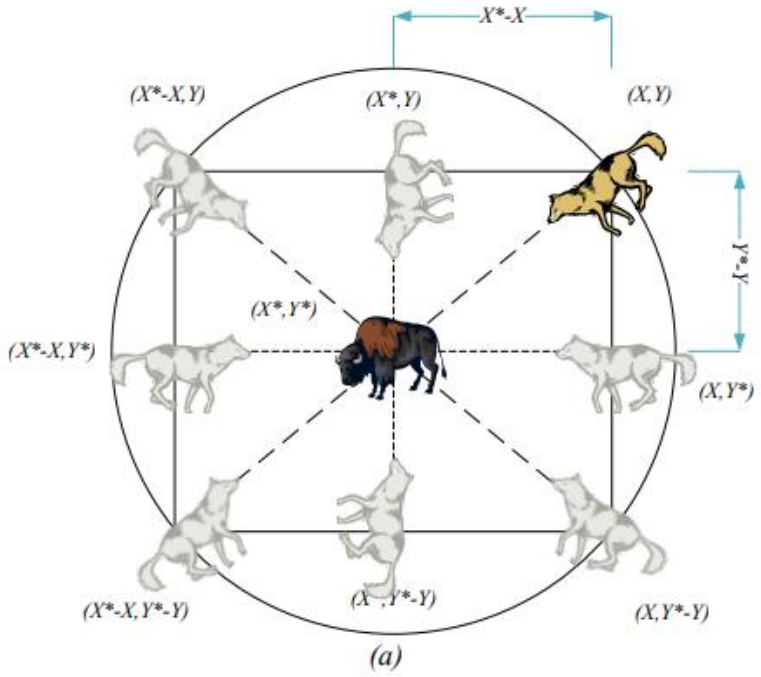

Рисунок 3.3 – Етап оточення здобичі вовком

3. Полювання

Полюванням зазвичай керують альфи, бети та дельти, які мають кращі знання про потенційне місце здобичі. Інші пошукові агенти повинні оновити свої позиції відповідно до кращих пошукових агентів. Оновлення позиції відбувається наступним чином:

$$\begin{aligned}\overline{X}_1 &= \overline{X}_\alpha - \overline{A}_1\left|\overline{C}_1\overline{X}_\alpha - \overline{X}\right| \\ \overline{X}_2 &= \overline{X}_\beta - \overline{A}_2\left|\overline{C}_2\overline{X}_\beta - \overline{X}\right|, \\ \overline{X}_3 &= \overline{X}_\delta - \overline{A}_3\left|\overline{C}_3\overline{X}_\delta - \overline{X}\right|\end{aligned}$$

$$\overline{X}(t+1) = \frac{\overline{X}_1 + \overline{X}_2 + \overline{X}_3}{3}. \qquad (3.3)$$

На рисунку 3.4 зображено, як пошуковий агент оновлює свою позицію відносно позицій альфа, бета та дельта вовків. Можна зауважити, що остаточна позиція буде у випадковому місці в межах кола, яке визначається позиціями альфи, бети та дельти у просторі пошуку. Іншими словами альфа, бета та дельта оцінюють позицію здобичі, а інші вовки оновлюють свої позиції випадково навколо здобичі.

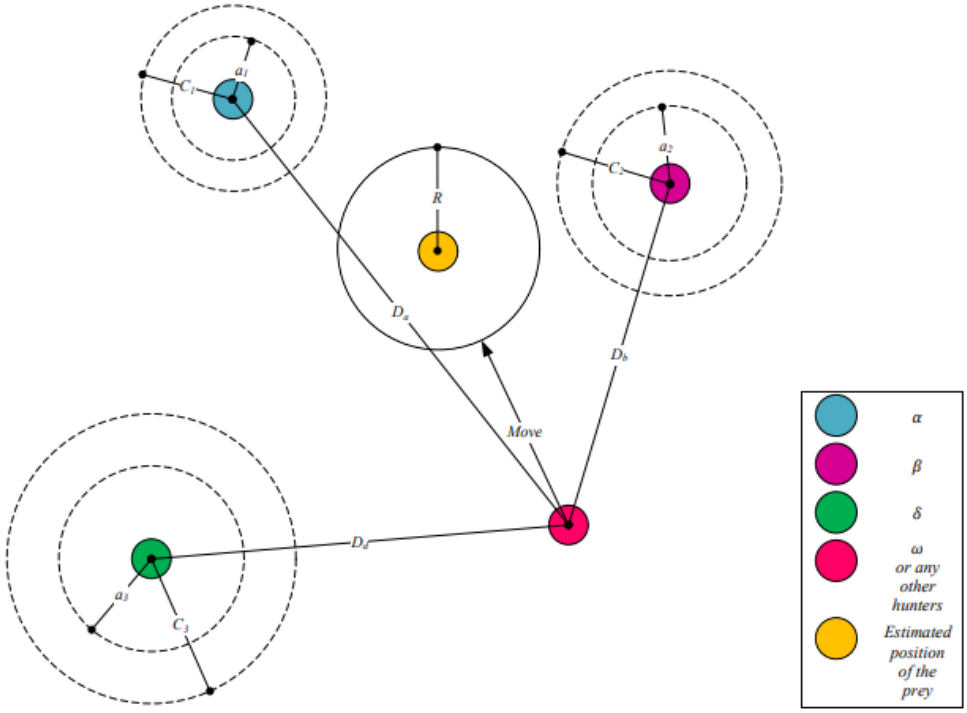

Рисунок 3.4 Схема зміни позиції пошукового агента

4. Напад на здобич.

Після завершення етапу полювання розпочинається новий етап – напад. На цьому етапі ключовим фактором впливу на зміну позиції агентів мають розташування альфа, бета та дельта агентів. Для математичного вираження процесу зближення моделі зі здобиччю використовуються два вектори $\bar{a}$ та $\bar{A}$. Параметр $\bar{a}$ лінійно зменшується від 2 до 0 від ітерації до ітерація, при цьому зменшуючи коливання $\bar{A}$. Іншими словами, $\bar{A}$ – це випадкове значення між $[-a, a]$. Коли випадкове значення $\bar{A}$ знаходиться між $[-1,1]$, наступне розташування агента буде між поточним значення та цільовою позицією (розташування здобичі) (рис. 3.5).

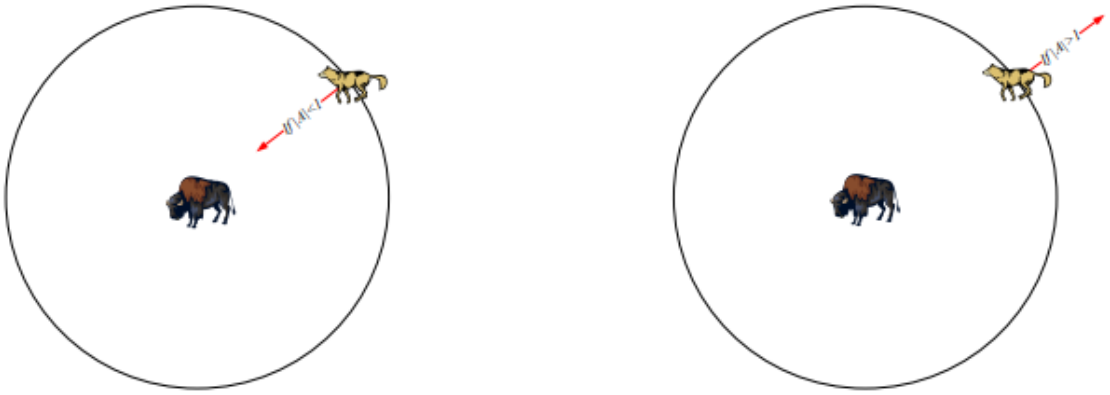

Рисунок 3.5 – Схематичне зображення атаки(а) та пошуку здобичі (б)

Використовуючи наведену вище схему оптимізації вовчою зграєю, сформуємо кроки алгоритму вовчої зграї для розв'язування поставлених задач.

Крок 1.Генерація популяції. Для популяції розміру M компоненти розв'язку (робочі потужності) генеруються випадковим чином. Для N генераторів i -й розв'язок X_i для задачі ЕРН виглядатиме наступним чином – $X_i = (X_{i1}, X_{i2}, ..., X_{iN})$ та для задачі ДРН – $X_i = (X_{i1}, X_{i2}, ..., X_{iN^*})$, $N^* = 24N$ при умові, що генератор працює 24 години.

Крок 2. Для кожного розв'язку X_i^k скорегувати його положення $X_{ij}^k = \begin{cases} X_{ij}^k, якщо\ X_{ij}^k \in \left[P_i^{\min} ... P_i^{\max}\right] \\ rand[P_i^{\min} ... P_i^{\max}], якщо\ X_{ij}^k \notin \left[P_i^{\min} ... P_i^{\max}\right] \end{cases}$.

Крок 3. Для кожного розв'язку X_i^k визначити фітнес-функцію за допомогою формули 2.8.

Крок 4. Відсортувати розв'язки у порядку зростання їх значень фітнес-функції. Присвоїти значення: $X_\alpha = X_1^k$, $X_\beta = X_2^k$, $X_\delta = X_3^k$.

Крок 5. Якщо досягнута максимальна кількість ітерацій – перейти до кроку 8.

Крок 6. Для кожного розв'язку (омега-вовки) $X_i^k, i = \left(\overline{4, M}\right)$, де M – кількість вовків. Визначити значення компонентів вектора розв'язку, використовуючи формулу 3.3.

Крок 7. Перейти до кроку 2.

Крок 8. Повернути розв'язок альфа-вовка X_α як найкращий знайдений розв'язок.

3.3 Конкретизація алгоритму вовчої зграї

Робота класичного АВЗ поділяється на дві стадії: пошуку та полювання (атаки здобичі) [49]. Перша стадія відповідає за глобальний пошук можливого оптимуму, другий же – за локальну оптимізацію. Після проведення дослідження особливостей роботи АВЗ при розв'язанні задачі ДРН, було виявлено важливу особливість – з огляду на велику кількість обмежень простір визначення цільової функції має велику кількість локальних оптимумів, через що алгоритм по завершенні першої стадії часто потрапляє в один із них і продовжує оптимізацію. Під час другого етапу отримати розв'язки, що істотно відрізнятимуться від домінуючих, неможливо, що значною мірою впливає на якість отриманого розв'язку. Для розв'язанні даної проблеми у роботі [48] запропоновано альтернативний спосіб визначення коефіцієнта соціальної складової, але цього не достатньо.

Як вже було описано, основною ідеєю вибору нових точок є поєднання впливу випадкової величини та існуючих трьох розв'язків, які із ростом кількості пройдених ітерацій збільшують свій вплив. Таким чином при переході до другого етапу усі згенеровані розв'язки будуть розміщені у околі домінуючих, і можлива область появи з кожною ітерацією звужуватиметься. У якості механізму виходу із локального оптимуму можна використати досить простий механізм – частина зграї буде не наближатись (атакувати) здобич а навпаки – віддалятись від неї (охороняти). Таким чином соціальна складова буде не звужувати область генерації нового розв'язку, а навпаки – віддаляти його від центру. Таким чином збільшується імовірність отримання нового локального оптимуму навіть під час другого етапу.

Підсумовуючи вищесказане, серед омега-вовків можна виділити дві групи – «мисливці» та «охоронці». Оскільки омега-вовки показують гірші результати, частина із них буде виконувати роль сторожі і досліджувати територію навколо зграї в пошуках нової здобичі.

Для визначення компонентів вектора розв'язку, що відповідають за «охоронців» будемо використовувати наступні формули:

$$\begin{aligned}
\overline{X}_1 &= \overline{X}_\alpha - \overline{A}_1 \left| \overline{C}_1 \overline{X}_\alpha - \frac{1}{2} \overline{X_i^k} \right| \\
\overline{X}_2 &= \overline{X}_\beta - \overline{A}_2 \left| \overline{C}_2 \overline{X}_\beta - \frac{1}{2} \overline{X_i^k} \right|, \\
\overline{X}_3 &= \overline{X}_\delta - \overline{A}_3 \left| \overline{C}_3 \overline{X}_\delta - \frac{1}{2} \overline{X_i^k} \right|
\end{aligned}$$

$$\overline{X_i^{k+1}} = \frac{\overline{X}_1 + \overline{X}_2 + \overline{X}_3}{3} \qquad (3.4)$$

Наведемо кроки роботи розробленого алгоритму.

Крок 1. Генерація популяції. Для популяції розміру M компоненти розв'язку (робочі потужності) генеруються випадковим чином. Для N генераторів i -й розв'язок X_i для задачі ЕРН виглядатиме наступним чином – $X_i = (X_{i1}, X_{i2}, ..., X_{iN})$

та для задачі ДРН – $X_i = (X_{i1}, X_{i2}, ..., X_{iN^*})$, $N^* = 24N$ при умові, що генератор працює 24 години.

Крок 2. Для кожного розв'язку X_i^k скорегувати його положення $X_{ij}^k = \begin{cases} X_{ij}^k, якщо\ X_{ij}^k \in [P_i^{\min} ... P_i^{\max}] \\ rand[P_i^{\min} ... P_i^{\max}],\ якщо\ X_{ij}^k \notin [P_i^{\min} ... P_i^{\max}] \end{cases}$.

Крок 3. Для кожного розв'язку X_i^k визначити фітнес-функцію за допомогою формули 2.8.

Крок 4. Відсортувати розв'язки у порядку зростання їх значень фітнес-функції. Присвоїти значення: $X_\alpha = X_1^k$, $X_\beta = X_2^k$, $X_\delta = X_3^k$.

Крок 5. Якщо досягнута максимальна кількість ітерацій – перейти до кроку 8.

Крок 6. Для кожного розв'язку вовка-мисливця $X_i^k, i = \overline{(4, Q)}$, де $Q = 0{,}7(M - 3)$, потрібно:

6.1 Визначити значення компонентів вектора розв'язку, використовуючи формулу 3.3.

Крок 7. Для кожного вовка-охоронця $X_i^k, i = \overline{(Q, M)}$:

7.1 Визначити значення компонентів вектора розв'язку, використовуючи формулу 3.4.

Крок 8. Перейти до кроку 2.

Крок 9.Повернути розв'язок альфа-вовка як найкращий знайдений розв'язок.

3.4 Генетичний алгоритм

Генетичні алгоритми (ГА, genetic algorithm) виникли на основі спостереження за природними процесами еволюції та селекції популяцій живих істот – особин певного виду – і моделювання їх принципів [55].

При описі ГА зазвичай використовують терміни, запозичені з генетики.

Популяція – це скінченна множина особин.

Особини, що входять до популяції, у ГА подаються хромосомами із закодованими в них множинами параметрів задачі, головним чином розв'язків, які інакше називаються *точками в просторі пошуку* (search points) задачі.

Хромосоми (інші назви – ланцюжки або кодові послідовності) – це впорядковані послідовності генів.

Ген (який також називається властивістю, знаком чи детектором) – це атомарний елемент генотипу, зокрема хромосоми.

Генотип, або структура – це набір хромосом даної особини.

Отже, особинами популяції можуть бути генотипи або одиничні хромосоми.

Фенотип – це сукупність зовнішніх і внутрішніх ознак, які відповідають даному генотипу, тобто декодована структура, або множина параметрів задачі (розв'язок, точка простору пошуку).

Алель – це альтернативне значення конкретного гена, також визначається як значення властивості або варіант властивості.

Локус (позиція) – указує на місце розміщення даного гена в хромосомі (ланцюжку амінокислот). Множина позицій генів – це локи.

Функція пристосованості (fitness function), яка ще називається *функцією корисності* чи *придатності,* виражає міру пристосованості особин у популяції. Ця функція дозволяє оцінити ступінь пристосованості конкретних особин у популяції й вибрати з них найбільш придатні відповідно до еволюційного принципу виживання найсильніших. У задачах оптимізації функція пристосованості зазвичай формується на основі цільової функції й оптимізується.

Процес появи в гені нових рис або підсилення вже наявних сильних рис, відображених у алелях, під впливом зовнішнього середовища називається *мутацією.* Основну схему еволюційного відбору, покладену в основу розробки ГА, наведено на рис. 3.6.

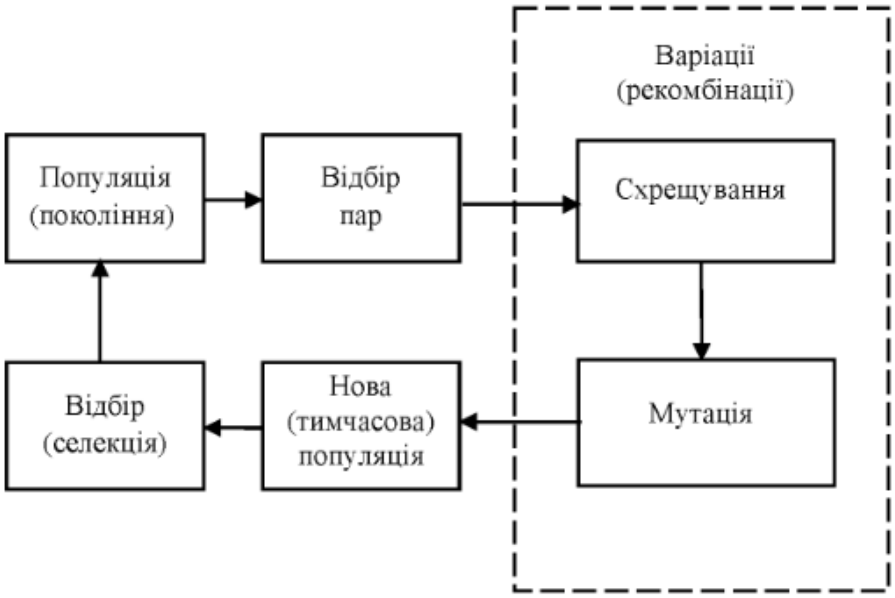

Рисунок 3.6 – Схема еволюційного відбору в ГА

Задача, яка потребує розв'язку, кодується таким чином, щоб її розв'язок можна було представити у вигляді масиву. Даний масив називається хромосомою. Популяція – набір хромосом на кожній ітерації. Кроссовер – операція утворення нової хромосоми на основі двох батьківських за допомогою поєднання частин їх вмісту за певними правилами. Мутація – випадкова зміна частини хромосоми.

Хромосома належить до даної схеми, якщо для кожної позиції j, $j = \overline{1, L}$, де L – довжина хромосоми; символ, що займає позицію j хромосоми, відповідає символу, що займає позицію j схеми, при чому символ * відповідає як 1, так і 0. Варто відмітити, що, якщо схема містить m символів *, то вона містить 2^m Хромосом, а кожна хромосома довжини L належить до 2^L схем.

Тепер розглянемо популяцію двійкових рядків довжини L. Імовірність проведення кроссовера дорівнює P_c. Тип кроссовера – 1-точковий. Нехай визначаюча довжина шаблона рівна $d(H)$, а його пристосування $f(H)$. Частка рядків, що відповідають шаблону H в поточному поколінні T, буде рівна $m(H,t)$. Нас буде цікавити частка рядків, що відповідають шаблону H, буде входити в популяцію наступного покоління – $m(H,t+1)$.

Розглянемо імовірність того, що кроссовер знищить уже існуючий шаблон. Очевидно, що якщо точка розриву не потрапить всередину існуючого шаблону, то

він не буде втрачений. Тобто, якщо $P_c \cdot d(H)/(L-1)$ - імовірність того, що точка розриву потрапить всередину шаблона, то $1-P_c \cdot d(H)/(L-1)$ - імовірність того, що кросове не знищить шаблон.

Згідно стратегії відбору ГА шанси особини прийняти участь в схрещуванні обраховуються як відношення: $\frac{f}{f_{ср}}$, де f - пристосування даної особини, а $f_{ср}$ - середнє пристосування. Таким чином імовірність того, що рядок, що відповідає шаблону H братиме участь в схрещуванні, буде дорівнювати $m(H,t) \cdot f(H,t) / f_{ср}(t)$. Взявши також до уваги імовірність знищення шаблону кроссовером, можна сформулювати теорему шаблонів Холланда:

$$\langle m(H,t+1) \rangle \geq m(H,t) \frac{f(H,t)}{\overline{f}(t)} \left[1 - p_c \frac{\delta(H)}{\ell - 1} \right],$$

Одним із основних недоліків даної теореми є те, що вона не враховує вплив мутації на створення і знищення шаблонів. Якщо вважати, що імовірність мутації одного розряду P_m, а порядок шаблона H дорівнює $o(H)$, то імовірність того, що мутація не зруйнує шаблон дорівнює $(1-P_m)^{o(H)}$. Тобто якщо розряд, що підлягає мутації, не потрапляє ні на одну фіксовану позицію всередині шаблона, то вона не змінюється. З урахуванням цього Холланд у 1995 році запропонував виправлену теорему шаблонів:

$$\langle m(H,t+1) \rangle \geq m(H,t) \frac{f(H,t)}{\overline{f}(t)} \left[1 - p_c \frac{\delta(H)}{\ell - 1} \right] (1-p_m)^{o(H)}$$

Генетичний алгоритм не гарантує отримання точного оптимуму задачі за скінченний час, але дає змогу отримати наближену відповідь при невеликих ресурсних затратах.

Введемо деякі позначення:

- m – кількість генераторів;
- L – масив ваг каменів;

– $cross1$ – шанс вибору кроссовера з однією точкою ($cross1 = \overline{0,100}$);

– $exchange$ – шанс мутації методом обміну генами ($exchange = \overline{0,100}$);

– n – величина популяції;

– p – кількість пар для розмноження;

– Lim – максимальна кількість ітерацій;

– out – кількість ітерацій без отримання нового оптимуму.

Загальну ідею генетичного алгоритму для задачі розподілу навантаження можна представити наступною схемою:

Крок 1. Задати значення вхідних параметрів.

Крок 2. Згенерувати $T[0]..T[n]$ хромосом із випадковими значеннями генів $Gen(T[i], j) \in [1; m], i = \overline{1, m}, j = \overline{1, n}$.

Крок 3. Проведення ітерацій алгоритму:

3.1. $count = out$.

3.2. **ЦИКЛ** від 0 до Lim:

3.2.1. Згенерувати $parents[0]...parents[p]$ ($0 < parents[i] \le m$) //двовимірний масив індексів батьків; чим менша $Fitness(T[i])$ тим більша імовірність індексу i потрапити у масив.

3.2.2. Ініціалізація порожньої популяції $tempT$.

3.2.3. **ЦИКЛ** по парам індексів (q, w) у $parents$ від 1 до p:

3.2.3.1. $children$ = $Multiplicate(T[q], T[w])$ // «спарюємо» 2 хромосоми та отримуємо 2 дітей.

3.2.3.2. $tempT = tempT \cup children$.

3.2.4. Ініціалізація масиву $similar$ довжини n та wor довжини m 0-ими значеннями .

3.2.5. **ЦИКЛ** по всім генам i від 1 до n.

3.2.5.1. **ЦИКЛ** по всім варіантам генів j від 1 до m.

3.2.5.1.1. $wor[Gen(T[j], i)] = wor[Gen(T[j], i)] + 1$.

3.2.5.2. $similar[i] = MAX(wor)$, задати значення 0 для wor .

3.2.6. Знайти кількість вбивств $kills = DeathNumber(AVER(similar)/m)$.

3.2.7. **ЦИКЛ** по i від $kills$ до 1:

3.2.7.1. $T = T \setminus T[ChooseDeadIndex()]$.

3.2.8. $T = T \cup tempT$.

3.2.9. Відсортувати T за зростанням $Fitness(T[i])$.

3.2.10. $T = T[1]..T[n]$.

3.3.**ЯКЩО** $best$ не ініціалізована **АБО** $best > Fitness(T[1])$ **ТО** $best = Fitness(T[1])$; $count = out$.

ІНАКШЕ $count = count - 1$.

Схема роботи *Multiplicate*

Параметри:

1. a,b - хромосоми для розмноження.

Крок 1. Визначити подібність хромосом $sim = Similarity(a,b)$.

Крок 2. Проводимо кросовер:

2.1. Ініціалізація масиву $children[1]..children[2]$ //дочірні хромосоми.

2.2. $i = 1$.

2.3. **ЯКЩО** $RANDOM(1000) < cross1*10$ **ТО:**

2.3.1. $crossP = RANDOM(n-2)+1$.

2.3.4. **ЦИКЛ** по всім перестановкам (x,y) з a,b

2.3.4.1. $Gens(children[i]) = \{Gen(x,0)..Gen(x,crossP)\}$

2.3.4.2. $i = i + 1$

2.4. **ІНАКШЕ**

2.4.1. $crossPs[1] = RANDOM(\|Gens(x)\| - 2)$

2.4.2. $crossPs[2] = RANDOM(\|Gens(x)\| - crossPs[1] - 1) + crossPs[1]$

2.4.3. **ЦИКЛ** по всім комбінація (x,y) з a,b

2.4.3.1. $Gens(children[i]) = \{Gen(x,0)..Gen(x,crossPs[1])\}$

2.4.3.2. $Gens(children[i]) = Gens(children[i]) \cup \{Gen(x, crossPs[2]+1)..Gen(x, x)\}$

Крок 3 Мутація дочірніх хромосом

3.1. $pos = PossFewMutations(sim)$ //визначити імовірність повторної мутації

3.2. **ЦИКЛ** по усіх хромосомам i з 1 до 2:

3.2.1. Ініціалізувати тимчасову хромосому $temp$.

3.2.2. $Gens(temp) = Gens(children[i])$.

3.2.3. **ЯКЩО** $RANDOM(100) > exchange$ **ТО:**

3.2.3.1. $Gen(temp, RANDOM(n) = RANDOM(m)$.

3.2.4. **ІНАКШЕ**

3.2.4.1. $first = RANDOM(n)$.

3.2.4.2. $next = RANDOM(n)$ // при цьому $first <> next$.

3.2.4.3. $SWAP(Gen(temp, first), Gen(temp, next))$.

3.2.5. **ЯКЩО** $Fitness(temp) < Fitness(T[n])$ **ТО** $Gens(children[i]) = Gens(temp)$.

3.2.6. **ЯКЩО** $RANDOM(1000000) < pos$ **ТО** повернутись до пункту 3.2.1.

Крок 4 Повернути масив $children$.

Допоміжні методи:

$SWAP(x, y)$ – міняє значення між змінними x і y.

$Gen(a, index)$ – повертає ген по номеру $index$ хромосоми a.

$Gens(a)$ – повертає масив генів хромосоми a.

$MAX(a)$ – повертає максимальне значення елементів масиву a.

$AVER(a)$ – повертає середнє значення елементів масиву a.

$ROUND(x)$ – повертає округлене значення x.

$RANDOM(x)$ – повертає випадкове ціле число від 1 до x.

3.5 Алгоритм зведення розв'язку до допустимого

З огляду на характер та кількість обмежень, отриманий розв'язок задачі із великою імовірністю не задовольнятиме усі обмеження, через що з'являється

необхідність у додатковому кроці по завершенню розв’язування, який би звів отриманий розв’язок до допустимого.

Усунення порушення заборонених зон

Насправді, уникати заборонені зони можна ще під час роботи самого алгоритму. Даного результату можна досягти скориставшись стандартним прийомом перетворення розв’язку від абсолютних величин до значень, що лежать на відрізку від 0 до 1, де 0 відповідає мінімально допустимій потужності і-го генератора, а 1 – максимальній. При цьому усі заборонені зони не входять у даний проміжок. За рахунок цього можна досягнути наступних ефектів:

– алгоритм пошуку стає нечутливим до масштабу значень робочих потужностей генераторів, заборонених зон і т.д.;

– під час пошуку алгоритм автоматично уникає усі заборонені зони, за рахунок чого збільшується точність;

– під час роботи алгоритму можна не враховувати штрафну функцію порушення заборонених зон, що спрощує обрахунок фітнес-функції.

Але при цьому варто взяти до уваги наступні недоліки:

– при кожному обрахунку фітнес-функції необхідно перетворювати відносні величини у абсолютні;

– на етапі «доопрацювання» розв’язку будуть необхідні абсолютні величини, через що з’являється можливість порушення обмеження заборонених зон.

Усунення порушення обмеження зміни швидкості робочої потужності

На жаль, метод, який би під час роботи алгоритму гарантував уникнення порушення обмежень даного виду і не погіршував би при цьому якість пошуку, не вдалось розробити. Через це усунення порушень даних обмежень необхідно буде виконати у якості етапу «доопрацювання».

Для розв’язування даної задачі можна застосувати методи динамічного програмування. За рахунок цього за скінченну кількість кроків можна отримати

розв’язок, який гарантовано не порушуватиме обмеження заборонених зон та зміни швидкості робочих потужностей.

Для цього розв’язок кожного генератора окремо розіб’ємо на підмасиви довжиною в 2 елементи. Щоб отримати даний результат скористаємось наступними правилами:

– поточний підмасив розбиваємо на підмасиви рівної довжини кратної 2;

– у випадку, коли довжина підмасивів не може одночасно задовольнити наведеним вище обмеженням одночасно – розбити правий підмасив на 1 елемент більше від запланованого.

Даний процес наведено на рисунку 3.7 для масиву із 12 елементів.

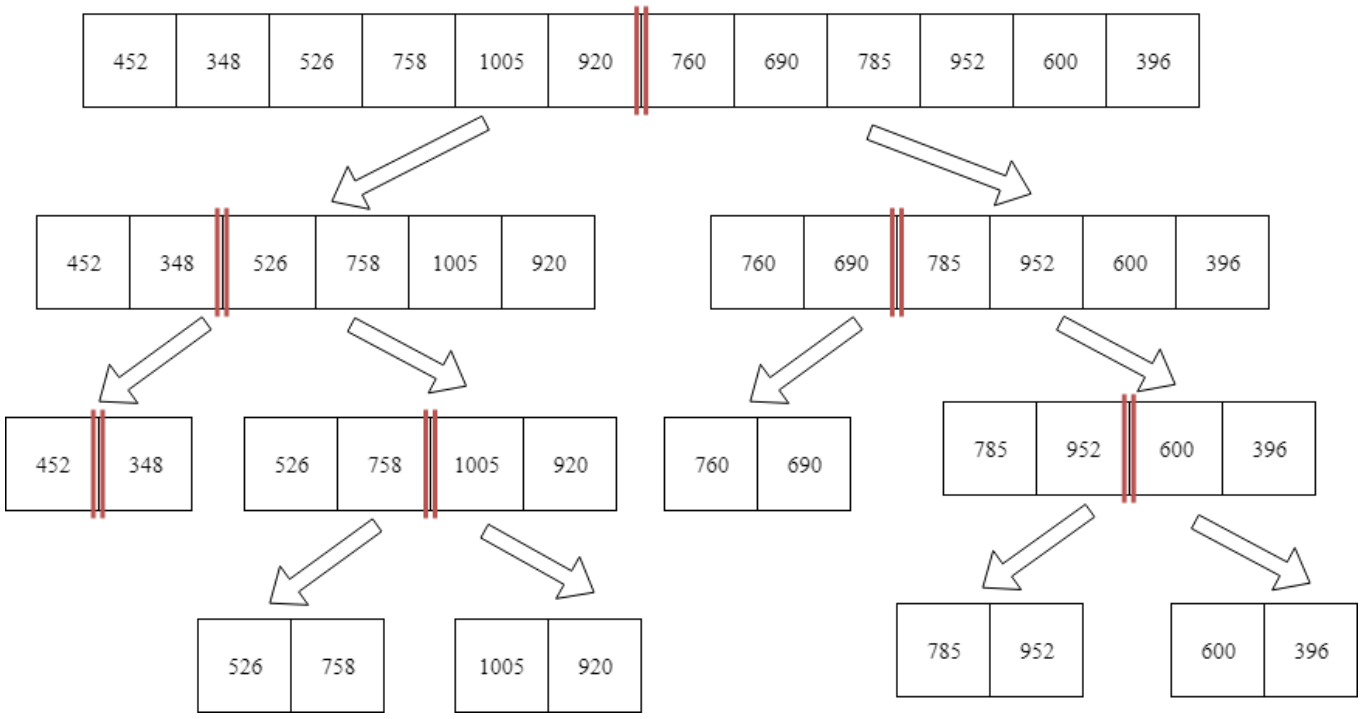

Рисунок 3.7 – Процес розбиття масиву на підмасиви

За рахунок даних кроків та факту, що початкова довжина є парним числом, а саме 24, гарантовано кінцеве розбиття складатиметься із масивів довжиною в 2 елементи.

На зворотному ході необхідно виконати зведення отриманих пар розв’язків довжиною в 2 елементи, а потім і підмасивів, що їх включають в себе. Таким чином подальші розв’язки базуватимуться на розв’язках менших задач.

При усуненні порушень обмежень на підмасиві довжиною 2 елементи можливі 5 випадків:

1) підмасив не порушує обмеження;

2) перший розв’язок більший за другий більше, ніж дозволяє обмеження зміни швидкості робочої потужності;

3) перший розв’язок більший за другий більше, ніж дозволяє обмеження зміни швидкості робочої потужності і, при його усуненні, другий розв’язок потрапляє у заборонену зону;

4) другий розв’язок більший за перший більше, ніж дозволяє обмеження зміни швидкості робочої потужності;

5) другий розв’язок більший за перший більше, ніж дозволяє обмеження зміни швидкості робочої потужності і, при його усуненні, перший розв’язок потрапляє у заборонену зону.

Для прикладу розглянемо підмасив із двох послідовних потужностей 220 та 120 МВт відповідно, заборонену зону 170-200 МВт та обмеження на швидкість зміни робочої потужності від -60 до +80 МВт. В результаті ми можемо збільшити значення другого розв’язку на 40 МВт і отримаємо допустимий розв’язок. Даний приклад подано на рисунку 3.8 і описує випадок 2.

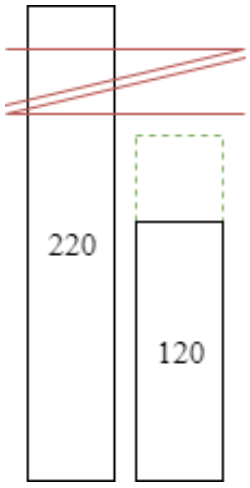

Рисунок 3.8 – Візуалізація другого випадку порушення обмеження

Формулювання даних випадків базується на одному правилі, що буде використано надалі в алгоритмі: при порушенні обмеження зміни швидкості робочої потужності або забороненої зони використовується лише оператор збільшення

одного із граничних значень (елементів підмасивів, що знаходяться на границі, або елементів підмасиву довжиною 2 елементи). За рахунок виконання даного правила отриманий в результаті розв’язок гарантовано буде задовольняти обмеження на попит і при цьому значно спрощує власну роботу.

Для випадку 3 збільшимо значення першого елементу підмасиву до 240 МВт. При збільшенні другого розв’язку для задоволення обмеження на швидкість зміни робочої потужності ми потрапимо у заборонену зону. Збільшивши розв’язок до значення верхньої межі забороненої зони, ми гарантовано отримаємо такий розв’язок, що задовольняє даним двом обмеженням. Графічне зображення даного випадку наведено на рисунку 3.9.

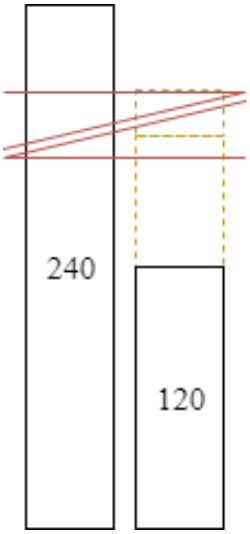

Рисунок 3.9 – Візуалізація третього випадку порушення обмеження

Третій та четвертий випадки аналогічні першому та другому випадкам відповідно.

Застосовуючи наведені вище правила, при зворотному ході, поєднуючи два підмасиви, у порівнянні беруть участь останній елемент першого підмасиву та перший другого. В результаті збільшення одного з елементів, отримується невизначеність задоволення обмежень у підмасиві, в який він входить. Для його усунення необхідно повторити виконання наведеного алгоритму для нього. Приклад порушення обмеження швидкості зміни робочої потужності наведено на рисунку 3.10.

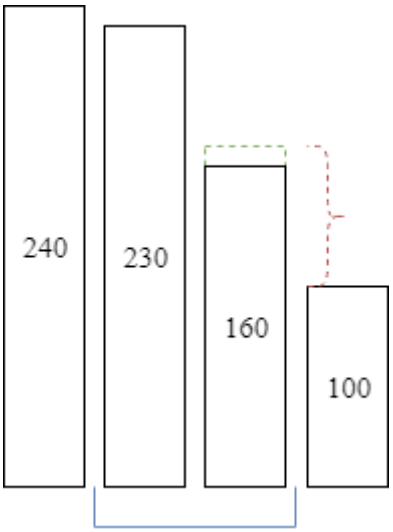

Рисунок 3.10 – Візуалізація зведення двох підмасивів

Наведемо розроблений алгоритм, сформулювавши набір процедур, необхідних для його роботи.

Процедура «Зведення розв'язку до допустимого»

Вхідні дані:

N – кількість генераторів;

M – розв'язок, що був отриманий після роботи основного алгоритму, поданий у вигляді масиву із розмірністю $24N$;

M_r – зведений розв'язок поданий у вигляді масиву із розмірністю $24N$.

Крок 1. Ініціалізувати порожній масив M_r

Крок 2. Для кожного розв'язку генератора із M:

Крок 2.1. Виконати процедуру *«Вкладеного зведення»* для поточного розв'язку генератора.

Крок 2.2. Додати зведений розв'язок до масиву M_r.

Крок 3. Повернути значення масиву M_r.

Процедура «Вкладеного зведення»

Вхідні дані:

M – розв'язок, що був отриманий після роботи основного алгоритму, поданий у вигляді масиву із розмірністю $24N$;

l_i – лівий індекс масиву;

r_i – правий індекс масиву.

Крок 1. Якщо виконується умова $r_i - l_i = 1$:

Крок 1.1. Викликати процедуру *«Одиничного зсуву»* для i-го генератора та повернути результат її виконання.

Крок 2. Інакше:

Крок 2.1. $c_i = (r_i + l_i)/2$ //цілочисельне ділення

Крок 2.2. Якщо $c_i - l_i = 2$, то $c_i = c_i - 1$.

Крок 2.3. Викликати процедуру *«Вкладеного зведення»* для i-го генератора, масиву M, l_i та c_i.

Крок 2.4. Викликати процедуру *«Вкладеного зведення»* для i-го генератора, масиву M, $c_i + 1$ та r_i.

Крок 2.3. Викликати процедуру *«Групового зведення»* для i-го генератора, масиву M, l_i, r_i та c_i.

Процедура «Одиничного зсуву»

Вхідні дані:

M – розв'язок, що був отриманий після роботи основного алгоритму, поданий у вигляді масиву із розмірністю $24N$;

l_i – лівий індекс масиву;

r_i – правий індекс масиву.

Крок 1. Присвоїти $r_v = M[r_i]$ та $l_v = M[l_i]$

Крок 2. Якщо $r_v - l_v < P_i^u$:

Крок 2.1. $M[r_i] = M[r_i] - (r_v - l_v + P_i^u)$.

Крок 2.2. Якщо $M[r_i]$ порушує обмеження заборонених зон:

Крок 2.2.1. $M[r_i] = \max\_brocken\_prohoboted\_zone + 0.0001$.

Крок 2.2.2. Викликати процедуру *«Одиничного зсуву»* для i-го генератора, масиву M, l_i та r_i повернути результати її виконання.

Крок 3. Якщо $r_v - l_v > generator\_\max\_ramp\_rate$:

Крок 3.1. $M[r_i] = M[r_i] + (r_v - l_v - P_i^u)$

Крок 3.2. Якщо $M[r_i]$ порушує обмеження заборонених зон:

Крок 3.2.1. $M[r_i] = \max\_brocken\_prohoboted\_zone + 0.0001.$

Крок 3.2.2. Викликати процедуру *«Одиничного зсуву»* для i-го генератора, масиву M, l_i та r_i

Процедура «Групового зсуву»

Вхідні дані:

M – розв'язок, що був отриманий після роботи основного алгоритму, поданий у вигляді масиву із розмірністю $24N$;

l_i – лівий індекс масиву;

r_i – правий індекс масиву;

c_i – проміжний індекс масиву.

Крок 1. Присвоїти $r_v = M[c_i + 1]$, $l_v = M[c_i]$.

Крок 2. Якщо $r_v - l_v < P_i^u$:

Крок 2.1. $M[r_i] = M[r_i] - (r_v - l_v + P_i^u)$.

Крок 2.2. Якщо $M[r_i]$ порушує обмеження заборонених зон:

Крок 2.2.1. $M[r_i] = \max\_brocken\_prohoboted\_zone + 0.0001$

Крок 2.2.2. Викликати процедуру *«Вкладеного зведення»* для i-го генератора, масиву M, c_{i+1} та r_i та повернути результати її виконання

Крок 3. Якщо $r_v - l_v > P_i^u$ то:

Крок 3.1. $M[r_i] = M[r_i] + (r_v - l_v - P_i^u)$

Крок 3.2. Якщо $M[r_i]$ порушує обмеження заборонених зон:

Крок 3.2.1. $M[r_i] = \max\_brocken\_prohoboted\_zone + 0.0001$

Крок 3.2.2. Викликати процедуру *«Одиничного зсуву»* для i-го генератора, масиву M, l_i та r_i та повернути результати її виконання

Висновки до розділу

Розділі подані три методи ройового інтелекту та генетичний алгоритм. Для кожного була наведена схема роботи алгоритму для розв’язування задач ЕРН та ДРН.

На основі класичного методу оптимізації вовчою зграєю було розроблено модифікований АВЗ. Головною особливістю модифікації являється боротьба із проблемою передвчасно збіжності. Ідея поділу зграї на «охоронців» та «мисливців» дозволяє не застрягати в локальних оптимумах.

Через велику кількість обмежень задачі розподілу навантаження розв’язки, що отримані після роботи основного алгоритму, часто не є допустимими. Було розроблено процедуру зведення отриманих розв’язків до допустимих після закінчення роботи основного алгоритму.

У частині графічного матеріалу подані блок-схеми розроблених алгоритмів ОРЧ, АВЗ та модифікації АВЗ, а також блок-схеми процедур зведення розв’язку до допустимого.

4 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Призначення та функції

Призначення розробленого програмного продукту автоматизувати процес знаходження ефективного розв'язку для задач ЕРН та ДРН.

Програмний продукт виконує наступні функції:

- зчитування файлів із вхідними даними;
- знаходження ефективного рішення за допомогою обраного алгоритму;
- проведення серій експериментів;
- генерація звіту у cvs форматі.

4.2 Середовище розробки та використані технології

Мовою для написання програмного продукту було вибрано **Ruby** з огляду на зручність та простоту створення гнучких систем. Веб-фреймворком було вибрано **Ruby on Rails** (як найкращий на сьогоднішній день фреймворк для Інтернет ресурсів, що створюються мовою Ruby. Інтегрованим середовищем розробки було обрано RubyMine.

Ruby – це мова динамічного програмування зі складною, але водночас виразною граматикою і базовою бібліотекою класів із чудовим та потужним API. Ruby ввібрав у себе особливості таких мов як Lisp, Smalltalk і Perl, але при цьому використовує граматику, якою без особливих складностей зможуть оволодіти програмісти, що пишуть на таких мовах як C/C++, C# та Java. Ruby є абсолютно об'єктно-орієнтованою мовою, але в ньому так само добре прижились і процедурні та функціональні стилі програмування. Він включає в себе потужні можливості для мета програмування [70].

Найважливішими перевагами Ruby є:

- присутні конструкції для обробки виключень, як в Java або Python, які дозволяють простіше працювати з помилками;

- поданий справжній mark-and-sweep збирач сміття для всіх Ruby об'єктів. Не потрібно вручну відстежувати кількість посилань в сторонніх бібліотеках;
- написання розширень на C в Ruby простіше ніж в Perl або Python за допомогою дуже елегантного API для виклику Ruby з C;
- є можливість довантажувати сторонні бібліотеки динамічно;
- реалізовані незалежні від операційної системи потоки. Таким чином, на будь-яких платформах, де ви запускаєте Ruby, ви також маєте можливість використовувати багатопоточність, незалежно від того чи підтримує дана система потоки чи ні;
- Ruby відрізняється високою переносимістю: він був розроблений здебільшого на GNU / Linux, але працює на багатьох типах UNIX, Mac OS X, Windows 95/98 / Me / NT / 2000 / XP / Vista / 7 / 8, DOS, BeOS, OS / 2 , і так далі [71].

RubyonRails (RoR) – веб-орієнтований фреймворк з відкритим кодом оптимізований для зручності написання програм і стійкою продуктивністю. Це надає можливість написання чудового коду, використовуючи погодження замість конфігурацій [72].

RoR – це повноцінний, багаторівневий фреймворк для побудови веб-застосувань, що використовують бази даних, який базується на архітектурі Модель-Представлення-Контролер (MVC).

В якості СУБД використано PostgreSQL – об’єктно-реляційна система управління базами даних, розробка якої триває з 1977. PostgreSQL вважається найдосконалішою СУБД, що поширюється з умовою відкритих висхідних кодів. В ній реалізована велика кількість можливостей, які зазвичай зустрічаються тільки у великих комерційних продуктах [73].

Сервер **PostgreSQL** написаний на мові C. Дана СУБД має багато переваг порівняно з іншими системами баз даних, а саме:

- покращена підтримка, ніж у пропрієтарній СУБД;
- значна економія на витрати персоналу;
- надзвичайно висока надійність і стабільність;
- масштабованість;

- кросплатформеність;
- призначена для задач, що потребують великої кількості пам’яті.

Vue.js – JavaScript-фреймворк, що використовує шаблон MVVM для створення інтерфейсів користувача на основі моделей даних, через реактивне зв'язування даних.

Одна із найвиразніших особливостей Vue – це ненав'язлива реактивна система. Моделі це просто плоскі JavaScript об'єкти. Це робить керування станами дуже простим та інтуїтивним. Vue надає оптимізований ре-рендеринг з коробки без потреби робити що-небудь додатково. Кожен компонент слідкує за своїми реактивними залежностями під час рендерингу, тому система знає точно коли має відбуватись ре-рендеринг і які компоненти потрібно ре-рендерити.

В якості проксі-сервера використовується Nginx (версії 1.8.0)–веб-сервер і поштовий проксі-сервер, що працює на Unix-подібних операційних системах. Nginx– простий, швидкий і надійний сервер, не переобтяжений функціями. Застосування Nginx доцільно насамперед для статичних веб-сайтів і як проксі-сервера перед динамічними сайтами [74]. За даними Netcraft на листопад 2014 року, число сайтів, що обслуговуються Nginx, перевищує 139 мільйонів, що робить його третім за популярністю веб-сервером у світі. Частка серед активних сайтів – 14,69%, що ставить Nginx на друге місце після веб-сервераApache .

За даними W3Techs, Nginx найбільш часто використовується на високонавантажених сайтах [75], займаючи перше місце за частотою використання серед 10 000 найбільш відвідуваних сайтів у світі – більше третини таких сайтів працює на Nginx.

В якості веб-сервера використано Unicorn – UnixRuby HTTP сервер для Rack застосувань. Він призначений для використання на Unix-подібних системах і використовує усі переваги та їх можливості. Unicorn багато поточний сервер і може працювати із будь-якою кількістю воркерів, при цьому він самостійно керує їх роботою, що спрощує роботу при розміщенні навантажених застосувань [76].

Для тестування було обрано RSpec – фреймворк для створення тестів для RoR застосувань. Він чудово реалізує принципи BDD і TDD. За рахунок інтуїтивно-зрозумілого функціоналу, час написання тестів значно скорочується [77].

4.3 Вимоги до технічного забезпечення

Вимоги до програмного забезпечення серверу:

- операційна система Debian 7.0 або вище;
- інтерпретатор Ruby 2.4.0;
- СУБД PostgreSQL 10.3.1;
- фреймворк Ruby on Rails 5.2;
- фреймворк Vue.js 2.5.16;
- бібліотека D3 5.1.0.
- бібліотека Sidekiq 4.1.2;
- веб-сервер nginx 1.8.0 або вище.

Вимоги до програмного забезпечення клієнта:

- браузер IE 10 або вище, Firefox, Chrome, Opera, Safari або будь-який інший браузер, що використовує один з наведених движків: Gecko 27.0 або вище, Trident 6.0 або вище, Webkit 537.31 або вище, Blink 537.11 або вище.

Мінімальні вимоги до технічного забезпечення:

- процесор – 1.6 Ггц, 1 ядро ЦП або краще;
- оперативна пам’ять не менш, ніж 1024 Мб;
- ПЗУ не менше, ніж 35 ГБ;
- доступу до мережі Інтернет (швидкість від 10 Мбіт/сек).

Усі користувачі взаємодіють з ресурсом за допомогою мережі Інтернет. Всі запити надходять до проксі-сервера Nginx, який перенаправляє їх до веб-сервера Unicorn. Веб-сервер виконує запит і при потребі взаємодіє із СУБД PostgreSQL. Структура мережі зображена на рисунку 4.1.

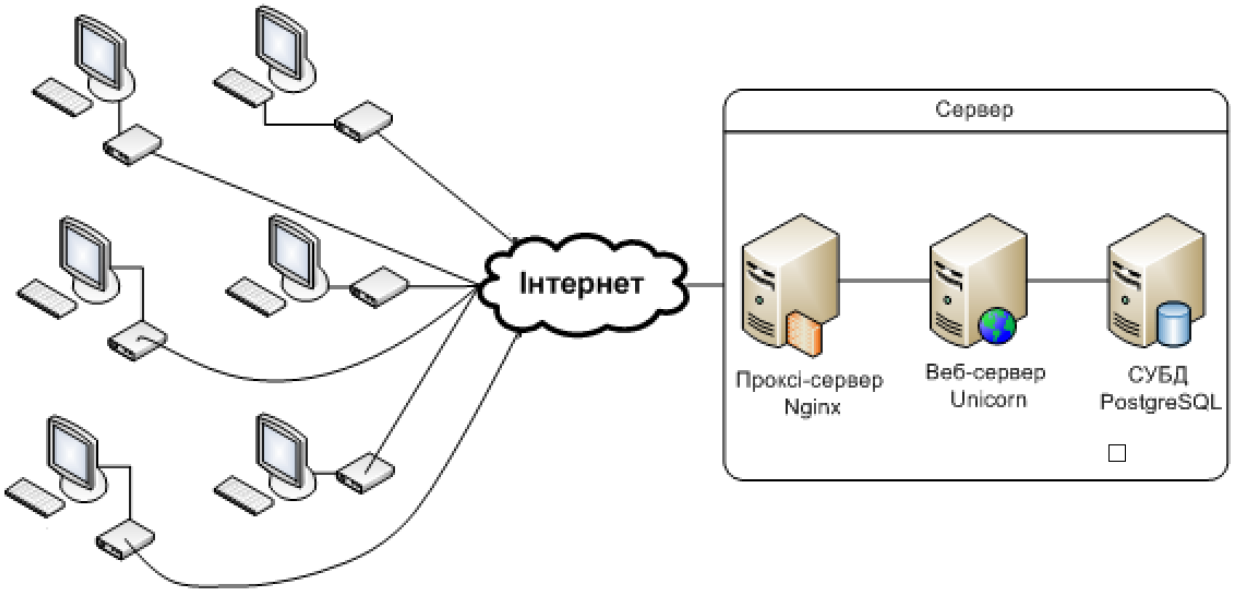

Рисунок 4.1 - Схема структурна мережі

4.4 Опис програмної реалізації

У частині графічного матеріалу подана схема структурна послідовності. На ній зображена типова послідовність дій, які виконуються при розв'язуванні задачі ЕРН або ДРН за допомогою розробленого застосування. Також визначено приналежність дій актора, необхідних для виконання поставленої задачі.

Також у частині графічного матеріалу подана схема структурна компонентів, на якій відображено компоненти, що використовуються у застосуванні, взаємозв'язки між ними та структурна схема класів програмного забезпечення, які представляють основні сутності WEB-ресурсу. До них входять:

– «Runner» – клас, що відповідає за додавання нової задачі в список на розв'язування. Задає кількість генераторів, кількість ітерації, алгоритм. По завершенню роботи алгоритму виводить результати.

– «Optimizer» – клас, що відповідає за запуск алгоритму ОРЧ, АВЗ, модифікацію АВЗ або ГА.

– «GWO» – клас, що відповідає за роботу АВЗ. Будує нову популяцію (вовчу зграю).

– «Pride» – клас, що представляє собою сутність «Зграя». Генерує новий розв’язок. Коректує значення альфа, бета, омега та дельта вовків. Сортує значення розв’язки після кожної ітерації. Відповідає за етап полювання на здобич (оновлює координати дельта-вовків).

– «WardenGWO» – клас, що відповідає за роботу модифікованого АВЗ. Будує нову популяцію (вовчу зграю).

– «WardenPride» – клас, що відвідає за створення популяції для модифікованого АВЗ. Ділить дельта-вовків на «охоронців» та «мисливців».

– «PSO» – клас, що відповідає за роботу алгоритму ОРЧ. Будує нову популяцію (рій частинок).

– «Particle» – клас, що представляє собою сутність «Чатинка». Генерує новий розв’язок. Обраховує швидкість частинки. Зберігає значення позиції найкращої частинки.

– «Generator» – клас, що представляє собою сутність «Генератор». Розраховує функцію вартість палива для одного генератора. Відповідає за генерацію робочої потужності для конкретного генератора та перевіряє чи згенерована потужність не виходить за межі.

– «ELD» – клас, що відповідає за представлення задачі ЕРН, містить в собі методи для розрахунку функції вартості палива та цільової функції. Знаходить значення загальних втрат за допомогою коефіцієнта та матриці втрат. Генерує початкову популяцію.

– «DELD» – клас, що відповідає за представлення задачі ДРН, містить в собі методи для розрахунку функції вартості палива, цільової функції. Знаходить значення загальних втрат за допомогою коефіцієнта та матриці втрат. Перевіряє попадання розв’язка в заборонені зони. Досліджує отриманий розв’язок.

– «UELD» – клас, що відповідає за зведену задачу. Розраховує фітнес функцію та оновлені обмеження зміни швидкості робочої потужності, заборонених зон та рівняння балансу.

4.5 Опис вхідних та вихідних даних

Вхідні дані

Всі вхідні дані користувач вводить в систему через web-інтерфейс або завантажує із файлу. До вхідних даних відносять:

– межі робочих потужностей;

– коефіцієнти функції витрат;

– обмеження швидкості переходу робочих потужностей (мінімальні та максимальні межі);

– заборонені зони роботи генератора;

– матриця, вектор та коефіцієнт втрат;

– попит на електроенергію (для задачі ЕРН за 1 годину, для задачі ДРН за 24 години);

– обрати необхідний алгоритм із списку.

Первісні дані зберігаються в базі даних. У таблицях 4.1-4.4 наведена структура таблиць бази даних.

Таблиця 4.1 – Структура таблиці «Генератори»

Назва таблиці	**Поля**	**Тип**	**Призначення**
generators	lower_limit	integer	Мінімальне значення робочої потужності для генератора
	upper_limit	integer	Максимальне значення робочої потужності для генератора
	a	float	Коефіцієнт функції витрат
	b	float	Коефіцієнт функції витрат
	c	float	Коефіцієнт функції витрат
	lower_ramp_rate	integer	Нижня межа зміни швидкості робочої потужності
	upper_ramp_rate	integer	Верхня межа зміни швидкості робочої потужності
	prohibited_zones	integer	Заборонені зони
	problem_id	integer	Зовнішній ключ
	created_at	datetime	Дата створення
	updated_at	datetime	Дата оновлення

Таблиця 4.3 – Структура таблиці «Розв’язки»

Назва таблиці	**Поля**	**Тип**	**Призначення**
solutions	task_id	integer	Зовнішній ключ
	stage	integer	Номер ітерації
	price	float	Вартість електроенергії на поточній ітерації. Визначається за допомогою цільової функції.
	penalty	float	Значення фітнес функції на поточній ітерації.
	mean_price	float	Середнє значення цільової функції
	mean_penalty	float	Середнє значення фітнес функції
	solution	float	Розв’язок
	created_at	datetime	Дата створення
	updated_at	datetime	Дата оновлення

Таблиця 4.2 – Структура таблиці «Задачі»

Назва таблиці	**Поля**	**Тип**	**Призначення**
problems	b_zero	float	Коефіцієнт витрат
	b_matrix	float	Матриця коефіцієнтів витрат
	scale	float	Масштаб
	power_demands	float	Робоча потужність, яку необхідно виробити
	name	string	Назва
	deleted	boolean	Показник, що визначає стан задачі
	created_at	datetime	Дата створення
	updated_at	datetime	Дата оновлення

Таблиця 4.4 – Структура таблиці «Завдання»

Назва таблиці	**Поля**	**Тип**	**Призначення**
tasks	max_iterations	integer	Максимальна кількість ітерацій
	max_agents	integer	Максимальна кількість агентів (вовків, частинок)
	problem_id	integer	Зовнішній ключ
	progress	integer	Кількість пройдених ітерацій
	created_at	datetime	Дата створення
	updated_at	datetime	Дата оновлення

Вихідні дані

На виході користувач отримує розв’язок задачі за допомогою обраного алгоритму. Перед виведенням на екран, отриманий основним алгоритмом розв’язок проходить етап «доопрацювання». На цьому кроці автоматично спрацьовує процедура зведення розв’язку до допустимого.

В результаті користувач отримує наступну інформацію:

– таблицю із необхідними робочими потужностями для кожної робочої години;

– графік залежності значень цільової функції від кількості ітерацій;

– графік залежності значень фітнес функції від кількості ітерацій;

– графік залежності значень функції штрафу від кількості ітерацій;

– загальна вартість палива.

Також у користувача є можливість завантажити звіт із більш детальним описом (рис. 4.2).

	A	B	C	D	E	F
1	Stage	Fitness function	Cost	Penalty	Mean Price	Mean Penalty
2	50	7470975626178	732132.981445312	7470974894045.1	703597.257215712	22402149018471
3	100	7470975626178	732132.981445312	7470974894045.1	698367.988769531	18791526393138.6
4	150	7470975626178	732132.981445312	7470974894045.1	697627.379557292	20120241506716.5
5	200	6917976911764	732560.889648438	6917976179203.56	701589.31515842	19650724643582.2
6	250	6917976911764	732560.889648438	6917976179203.56	708378.212076823	17206520000624.1
7	300	6917976911764	732560.889648438	6917976179203.56	707257.910970052	18390959842663.2
8	350	6917976911764	732560.889648438	6917976179203.56	707977.939832899	18261236825009.6
9	400	6272629228347	738168.0390625	6272628490179.05	706222.807779948	18164988413896.3
10	450	6272629228347	738168.0390625	6272628490179.05	710442.723253038	17140151572807.7
11	500	6272629228347	738168.0390625	6272628490179.05	709598.004286024	16311724590820.6
12	550	5672429759074	734536.616210938	5672429024537.83	708197.326497396	16426875296742.2

Рисунок 4.2 – Приклад звіту із детальним описом

Висновки до розділу

У даному розділі описано засоби розробки. Наведено опис обраної мови програмування, web -фреймворк для неї, проксі-сервер, web-сервер та СУБД, надано їх основну характеристику та переваги. Наведено короткий опис фреймворку для проведення тестування. Наведено вимоги до програмного забезпечення серверу та

клієнта. Описано мінімально вимоги до технічного забезпечення серверу. Виконано опис обчислювальної мережі та наведено її структуру на рисунку 4.1.

Наведено опис головних класів системи та їх схематичне зображення, яке наведено у додатку графічного матеріалу. Додано опис полів таблиць бази даних. Наведено опис вхідних та вихідних даних.

Програмно реалізовано ГА, ОРЧ, АВЗ та модифікація АВЗ. Також реалізовано процедури для зведення розв’язків до допустимих для обмеження швидкості зміни робочої потужності генератора та обмеження на заборонені зони.

Програмне забезпечення дозволяє ввести вхідні дані за допомогою web-інтерфейсу та за допомогою зчитування файлу. У користувач є можливість обрати алгоритм, яким він бажає розв’язувати задачу. Всі розв’язки в кінці роботи основного алгоритму зводяться до допустимих. Результати виводяться на екран, а для більш детальної інформації можна завантажити звіт у форматі .csv.

5 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ АЛГОРИТМІВ

5.1 Процес проведення експериментів

Експерименти виконуються для порівняння ефективності роботи реалізованих алгоритмів та перевірки процедури зведення отриманих розв'язків до допустимих.

Для проведення першої частини експерименту опишемо вхідні параметри для задач ЕРН та ДРН, параметри алгоритмів, значення коефіцієнтів фітнес функції та характеристики апаратного забезпечення.

Вхідні параметри для задач ЕРН та ДРН

Для проведення експериментів були використані два набори вхідних параметрів для задач ЕРН та ДРН. На рисунку 5.1 представлені матриця, вектор та коефіцієнт втрат для першого набору, що описує роботу електромережі із 15 генераторів [23].

$$B_{ij} = 10^{-3} \cdot \begin{bmatrix} 1.4 & 1.2 & 0.7 & -0.1 & -0.3 & -0.1 & -0.1 & -0.1 & -0.3 & -0.5 & -0.3 & -0.2 & 0.4 & 0.3 & -0.1 \\ 1.2 & 1.5 & 1.3 & 0.0 & -0.5 & -0.2 & 0.0 & 0.1 & -0.2 & -0.4 & -0.4 & 0.0 & 0.4 & 1.0 & -0.2 \\ 0.7 & 1.3 & 7.6 & -0.1 & -1.3 & -0.9 & -0.1 & 0.0 & -0.8 & -1.2 & -1.7 & 0.0 & -2.6 & 11.1 & -2.8 \\ -0.1 & 0.0 & -0.1 & 3.4 & -0.7 & -0.4 & 1.1 & 5.0 & 2.9 & 3.2 & -1.1 & 0.0 & 0.1 & 0.1 & -2.6 \\ -0.3 & -0.5 & -1.3 & -0.7 & 9.0 & 1.4 & -0.3 & -1.2 & -1.0 & -1.3 & 0.7 & -0.2 & -0.2 & -2.4 & -0.3 \\ -0.1 & -0.2 & -0.9 & -0.4 & 1.4 & 1.6 & 0.0 & -0.6 & -0.5 & -0.8 & 1.1 & -0.1 & -0.2 & -1.7 & 0.3 \\ -0.1 & 0.0 & -0.1 & 1.1 & -0.3 & 0.0 & 1.5 & 1.7 & 1.5 & 0.9 & -0.5 & 0.7 & 0.0 & -0.2 & -0.8 \\ -0.1 & 0.1 & 0.0 & 5.0 & -1.2 & -0.6 & 1.7 & 16.8 & 8.2 & 7.9 & -2.3 & -3.6 & 0.1 & 0.5 & -7.8 \\ -0.3 & -0.2 & -0.8 & 2.9 & -1.0 & -0.5 & 1.5 & 8.2 & 12.9 & 11.6 & -2.1 & -2.5 & 0.7 & -1.2 & -7.2 \\ -0.5 & -0.4 & -1.2 & 3.2 & -1.3 & -0.8 & 0.9 & 7.9 & 11.6 & 20.0 & -2.7 & -3.4 & 0.9 & -1.1 & -8.8 \\ -0.3 & -0.4 & -1.7 & -1.1 & 0.7 & 1.1 & -0.5 & -2.3 & -2.1 & -2.7 & 14.0 & 0.1 & 0.4 & -3.8 & 16.8 \\ -0.2 & 0.0 & 0.0 & 0.0 & -0.2 & -0.1 & 0.7 & -3.6 & -2.5 & -3.4 & 0.1 & 5.4 & -0.1 & -0.4 & 2.8 \\ 0.4 & 0.4 & -2.6 & 0.1 & -0.2 & -0.2 & 0.0 & 0.1 & 0.7 & 0.9 & 0.4 & -0.1 & 10.3 & -10.1 & 2.8 \\ 0.3 & 1.0 & 11.1 & 0.1 & -2.4 & -1.7 & -0.2 & 0.5 & -1.2 & -1.1 & -3.8 & -0.4 & -10.1 & 57.8 & -9.4 \\ -0.1 & -0.2 & -2.8 & -2.6 & -0.3 & 0.3 & -0.8 & -7.8 & -7.2 & -8.8 & 16.8 & 2.8 & 2.8 & -9.4 & 128.3 \end{bmatrix}$$

$$B_{i0} = 10^{-3} \cdot [-0.1 \quad -0.2 \quad 2.8 \quad -0.1 \quad 0.1 \quad -0.3 \quad -0.2 \quad -0.2 \quad 0.6 \quad 3.9 \quad -1.7 \quad 0.0 \quad -3.2 \quad 6.7 \quad -6.4]$$

$$B_{00} = 0.0055$$

Рисунок 5.1 – Перший набір вхідних даних: матриця, вектор та коефіцієнт втрат електромережі

У таблиці 5.1 наведені мінімальні та максимальні межі робочих потужностей, коефіцієнти функції витрат, обмеження швидкості переходу робочих потужностей та заборонені зони для кожного генератора для першої задачі.

Таблиця 5.1 – Перший набір вхідних даних: межі робочих потужностей, коефіцієнти функції витрат, обмеження швидкості переходу робочих потужностей та заборонені зони для кожного генератора для першої задачі.

№	Межі робочих потужностей		Коефіцієнти функції витрат			Обмеження швидкості переходу робочих потужностей	Заборонені зони
	$P_i^{\min}$	$P_i^{\max}$	a	b	c		
1	150	455	0.000299	10.1	671	[80, 120]	[185..255, 305..335, 420..450]
2	150	455	0.000183	10.2	574	[80, 120]	
3	20	130	0.001126	8.8	374	[130, 130]	
4	20	130	0.001126	8.8	374	[130, 130]	
5	150	470	0.000205	10.4	461	[80, 120]	[180..200, 305..335, 390..420]
6	135	460	0.000301	10.1	630	[80, 120]	[230..255, 365..395, 430..455]
7	155	465	0.000364	9.8	548	[80, 120]	
8	60	300	0.000338	11.2	227	[65, 100]	
9	25	162	0.000807	11.2	173	[60, 100]	
10	25	160	0.001203	10.7	175	[60, 100]	
11	20	80	0.003586	10.2	186	[80, 80]	[30..40, 55..65]
12	20	80	0.005513	9.9	230	[80, 80]	
13	25	85	0.000371	13.1	225	[80, 80]	
14	15	55	0.001929	12.1	309	[55, 55]	[60..73, 82..95, 125..130]
15	15	55	0.004447	12.4	323	[55, 55]	

З огляду на особливість задачі ДРН необхідно окремо розглядати зміну попиту, що відбувається в продовж доби. Тому для першого набору вхідних даних для задачі ДРН в таблиці 5.2 подані значення загального попиту на електроенергію, що відповідають попиту за кожну годину.

Таблиця 5.2 – Перший набір вхідних даних: попит на електроенергію

Година	Попит	Година	Попит	Година	Попит	Година	Попит
1	1600	7	2700	13	1900	19	2630
2	1681	8	2690	14	2100	20	2930
3	1610	9	2000	15	2200	21	2830
4	1590	10	2100	16	2095	22	2730
5	1800	11	2010	17	2600	23	2230
6	2300	12	2100	18	2200	24	1890

На рисунку 5.2 зображена матриця втрат електромережі, що складається із 20 генераторів.

$B_{ij} = 10^{-3} \cdot$

$$\begin{bmatrix}
8.70 & 0.43 & -4.61 & 0.36 & 0.32 & -0.66 & 0.96 & -1.60 & 0.80 & -0.10 & 3.60 & 0.64 & 0.79 & 2.10 & 1.70 & 0.80 & -3.20 & 0.70 & 0.48 & -0.70 \\
0.43 & 8.30 & -0.97 & 0.22 & 0.75 & -0.28 & 5.04 & 1.70 & 0.54 & 7.20 & -0.28 & 0.98 & -0.46 & 1.30 & 0.80 & -0.20 & 0.52 & -1.70 & 0.80 & 0.20 \\
-4.61 & -0.97 & 9.00 & -2.00 & 0.63 & 3.00 & 1.70 & -4.30 & 3.10 & -2.00 & 0.70 & -0.77 & 0.93 & 4.60 & -0.30 & 4.20 & 0.38 & 0.70 & -2.00 & 3.60 \\
0.36 & 0.22 & -2.00 & 5.30 & 0.47 & 2.62 & -1.96 & 2.10 & 0.67 & 1.80 & -0.45 & 0.92 & 2.40 & 7.60 & -0.20 & 0.70 & -1.00 & 0.86 & 1.60 & 0.87 \\
0.32 & 0.75 & 0.63 & 0.47 & 8.60 & -0.80 & 0.37 & 0.72 & -0.90 & 0.69 & 1.80 & 4.30 & -2.80 & -0.70 & 2.30 & 3.60 & 0.80 & 0.20 & -3.00 & 0.50 \\
-0.66 & -0.28 & 3.00 & 2.62 & -0.80 & 11.8 & -4.90 & 0.30 & 3.00 & -3.00 & 0.40 & 0.78 & 6.40 & 2.60 & -0.20 & 2.10 & -0.40 & 2.30 & 1.60 & -2.10 \\
0.96 & 5.04 & 1.70 & -1.96 & 0.37 & -4.90 & 8.24 & -0.90 & 5.90 & -0.60 & 8.50 & -0.83 & 7.20 & 4.80 & -0.90 & -0.10 & 1.30 & 0.76 & 1.90 & 1.30 \\
-1.60 & 1.70 & -4.30 & 2.10 & 0.72 & 0.30 & -0.90 & 1.20 & -0.96 & 0.56 & 1.60 & 0.80 & -0.40 & 0.23 & 0.75 & -0.56 & 0.80 & -0.30 & 5.30 & 0.80 \\
0.80 & 0.54 & 3.10 & 0.67 & -0.90 & 3.00 & 5.90 & -0.96 & 0.93 & -0.30 & 6.50 & 2.30 & 2.60 & 0.58 & -0.10 & 0.23 & -0.30 & 1.50 & 0.74 & 0.70 \\
-0.10 & 7.20 & -2.00 & 1.80 & 0.69 & -3.00 & -0.60 & 0.56 & -0.30 & 0.99 & -6.60 & 3.90 & 2.30 & -0.30 & 2.80 & -0.80 & 0.38 & 1.90 & 0.47 & -0.26 \\
3.60 & -0.28 & 0.70 & -0.45 & 1.80 & 0.40 & 8.50 & 1.60 & 6.50 & -6.60 & 10.7 & 5.30 & -0.60 & 0.70 & 1.90 & -2.60 & 0.93 & -0.60 & 3.80 & -1.50 \\
0.64 & 0.98 & -0.77 & 0.92 & 4.30 & 0.78 & -0.83 & 0.80 & 2.30 & 3.90 & 5.30 & 8.00 & 0.90 & 2.10 & -0.70 & 5.70 & 5.40 & 1.50 & 0.70 & 0.10 \\
0.79 & -0.46 & 0.93 & 2.40 & -2.80 & 6.40 & 7.20 & -0.40 & 2.60 & 2.30 & -0.60 & 0.90 & 11.0 & 0.87 & -1.00 & 3.60 & 0.46 & -0.90 & 0.60 & 1.50 \\
2.10 & 1.30 & 4.60 & 7.60 & -0.70 & 2.60 & 4.80 & 0.23 & 0.58 & -0.30 & 0.70 & 2.10 & 0.87 & 3.80 & 0.50 & -0.70 & 1.90 & 2.30 & -0.97 & 0.90 \\
1.70 & 0.80 & -0.30 & -0.20 & 2.30 & -0.20 & -0.90 & 0.75 & -0.10 & 2.80 & 1.90 & -0.70 & -1.00 & 0.50 & 11.0 & 1.90 & -0.80 & 2.60 & 2.30 & -0.10 \\
0.80 & -0.20 & 4.20 & 0.70 & 3.60 & 2.10 & -0.10 & -0.56 & 0.23 & -0.80 & -2.60 & 5.70 & 3.60 & -0.70 & 1.90 & 10.8 & 2.50 & -1.80 & 0.90 & -2.60 \\
-3.20 & 0.52 & 0.38 & -1.00 & 0.80 & -0.40 & 1.30 & 0.80 & -0.30 & 0.38 & 0.93 & 5.40 & 0.46 & 1.90 & -0.80 & 2.50 & 8.70 & 4.20 & -0.30 & 0.68 \\
0.70 & -1.70 & 0.70 & 0.86 & 0.20 & 2.30 & 0.76 & -0.30 & 1.50 & 1.90 & -0.60 & 1.50 & -0.90 & 2.30 & 2.60 & -1.80 & 4.20 & 2.20 & 0.16 & -0.30 \\
0.48 & 0.80 & -2.00 & 1.60 & -3.00 & 1.60 & 1.90 & 5.30 & 0.74 & 0.47 & 3.80 & 0.70 & 0.60 & -0.97 & 2.30 & 0.90 & -0.30 & 0.16 & 7.60 & 0.69 \\
-0.70 & 0.20 & 3.60 & 0.87 & 0.50 & -2.10 & 1.30 & 0.80 & 0.70 & -0.26 & -1.50 & 0.10 & 1.50 & 0.90 & -0.10 & -2.60 & 0.68 & -0.30 & 0.69 & 7.00
\end{bmatrix}$$

Рисунок 5.2 – Задача 2: матриця втрат електромережі

В таблиці 5.3 подані дані про межі робочих потужностей, коефіцієнти функції витрат, обмеження швидкості переходу робочих потужностей та заборонені зони для другої задачі, що складається із 20 генераторів.

Таблиця 5.3 – Другий набір вхідних даних: межі робочих потужностей, коефіцієнти функції витрат, обмеження швидкості переходу робочих потужностей та заборонені зони для кожного генератора для першої задачі.

№	Межі робочих потужностей		Коефіцієнти функції витрат			Обмеження швидкості переходу робочих потужностей	Заборонені зони
	$P_i^{\min}$	$P_i^{\max}$	a	b	c		
1	150	600	0.00068	18.19	1000	[80, 120]	[185..255, 305..335, 420..450]
2	50	200	0.00071	19.26	970	[80, 120]	
3	50	200	0.0065	19.8	600	[130, 130]	
4	50	200	0.005	19.1	700	[130, 130]	
5	50	160	0.00738	18.1	420	[80..100, 120..140]	
6	20	100	0.00612	19.26	360	[50..60]	
7	25	125	0.0079	17.14	490	[80, 120]	[80..100, 120..140]
8	50	150	0.00813	18.92	660	[65, 100]	[50..60, 70..80]
9	50	200	0.00522	18.27	765	[60, 100]	
10	30	150	0.00573	18.92	770	[65, 100]	
11	100	300	0.00480	16.69	800	[80, 80]	
12	150	500	0.0031	16.76	970	[80, 80]	
13	40	160	0.0085	17.36	900	[80, 80]	
14	20	130	0.00511	18.7	700	[55, 55]	[60..65, 80..91, 105..120]
15	15	185	0.00398	18.7	450	[55, 55]	
16	20	80	0.0712	14.26	370	[55, 55]	
17	30	85	0.0089	19.14	480	[55, 55]	
18	30	120	0.00713	18.92	680	[55, 55]	
19	40	120	0.00622	18.47	700	[55, 55]	[30..35, 45..55, 75..80, 98..100]
20	30	100	0.00773	19.79	850	[55, 55]	

Для другого варіанту задачі ДРН в таблиці 5.4 наведені значення загального попиту на електроенергію, що відповідають попиту за кожну годину.

Таблиця 5.4 – Задача 2: попит на електроенергію

Година	**Попит**	**Година**	**Попит**	**Година**	**Попит**	**Година**	**Попит**
1	1500	7	1700	13	1800	19	2830
2	1581	8	2200	14	2000	20	2730
3	1510	9	2600	15	1800	21	2630
4	1490	10	2590	16	2000	22	2630
5	1700	11	1900	17	2100	23	2130
6	2200	12	2000	18	2530	24	1790

Значення коефіцієнтів фітнес функції

$$q_1 = 1,$$

$$q_2 = 50,$$

$$q_3 = 50.$$

Характеристики апаратного забезпечення

Експерименти проводилися на комп’ютері з такими характеристиками:

- обсяг ОЗУ: 8 Гб;
- тактова частота процесора: 2.9 ГГц;
- двох ядерний процесор Intel Core i5;
- тактова частота процесора 2.9 ГГц.

Параметри алгоритмів

Для алгоритму ОРЧ при проведені експериментів використано наступні значення параметрів.

- коефіцієнт когнітивної складової:

$$c_1 = 1;$$

- коефіцієнт соціальної складової:

$$c_2 = 2;$$

- вага інерції:

$$w = 0{,}5;$$

- розмір популяції (кількість часток у рої):

$$N_D = 50.$$

Опишемо значення параметрів для ГА:

– розмір популяції:

$$N = 50;$$

– кількість пар для розмноження:

$$q = 4;$$

– ймовірність першочергового вибору домінанта:

$$d = 0{,}3;$$

– ймовірність мутації:

$$m = 0{,}25;$$

– ймовірність кроссовера:

$$k = 0{,}45.$$

Для класичного та модифікованого АВЗ встановлено наступні вхідні параметри:

– вектори випадкових величин:

$$r_1 = rand(0{,}1);$$

$$r_2 = rand(0{,}1);$$

– вектор, компоненти якого лінійно зменшуються:

$$\bar{a} = 2(1 - i / \max\_iterations),$$

де i – поточна ітерація, $\max\_iterations$ – максимальна кількість ітерацій;

– вектор соціальних коефіцієнтів:

$$\bar{A} = 2\bar{a}\bar{r}_1 - \bar{a};$$

– вектор когнітивних коефіцієнтів:

$$\bar{C} = 2\bar{r}_2;$$

– розмір популяції (кількість вовків у зграї):

$$M = 50.$$

Відсоток «охоронців» у зграї для модифікованого АВЗ був вибраний експериментальним шляхом. Дослідження проводилися для задачі ДРН із системою, що складається із 15 генераторів. На початку роботи алгоритму випадковим чином

був обраний відсоток «охоронців» у вказаних межах. Для кожного випадку було проведено 50 тестових запусків. Результати роботи алгоритму були усереднені для кожного випадку та були порівняні із результатом АВЗ. Різниця між значеннями фітнес функції модифікованого АВЗ відносно АВЗ обчислюється за формулою

$$\Delta = \frac{f_{МАВЗ} - f_{АВЗ}}{f_{АВЗ}},$$

де $f_{МАВЗ}$ – це усереднене значення фітнес функції модифікованого АВЗ після 50 тестових запусків, $f_{АВЗ}$ – це усереднене значення фітнес функції АВЗ після 50 тестових запусків.

У таблиці 5.5 наведено залежність значень фітнес функції від кількості «охоронців» у грaї.

Таблиця 5.5 – Залежність значень фітнес функції від кількості «охоронців» у грaї.

Відсоток «охоронців» у грaї	0-5%	5-15%	15-30%	30-50%	50-75%
Значення фітнес функції	494555699074	469300965307	449140322071	499541930162	554422729165
Коефіцієнт ефективності відносно результатів АВЗ	-0,01877	-0,068177	-0,10888	-0,00828	0,10021

Спираючись на результати, що подані в таблиці 5.5, можна зробити висновок, що кількість «охоронців» у грaї має бути в мережах від 15% до 30% від загальної кількості вовків. Тому кількість «охоронців» буде обчислюватися наступною формулою:

$$k = rand(15,30).$$

5.2 Порівняння ефективності роботи алгоритмів

Для перевірки ефективності роботи реалізованих алгоритмів проведено два обчислювальні експерименти:

1. Визначення математичного сподівання і дисперсії значень цільової функції і фітнес функції в точці розв’язку.

2. Дослідження тенденції зміни значення цільової функції і фітнес функції всіх алгоритмів протягом процесу оптимізації.

Зауважимо, що для кожного тестового запуску було згенеровано нову популяцію. Для першого експерименту було виконано по 100 тестових запусків кожного алгоритму для кожної кількості ітерацій.

В таблицях 5.6-5.9 наведено порівняння математичного очікування та дисперсії вартості палива відносно значень цільової функції та фітнес функції для обраних алгоритмів для задачі ЕРН із системою, що складається із 15 генераторів.

Таблиця 5.6 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих ГА в залежності від максимальної кількості ітерацій для задачі ЕРН (15 генераторів).

Ітерація	**ГА**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	39128,78	1030510,05	907,7065	315616,74
250	38824,42	1024139,91	1773,275	358438,02
500	38983,42	1111947,78	2173,693	394176,94
1000	39460,3	2134414,78	1651,57	2349456,67
1500	3897,78	925450,46	2179,447	93421,12
2000	39278,14	945093,85	3633,555	3933206,09

Таблиця 5.7 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих алгоритмом ОРЧ в залежності від максимальної кількості ітерацій для задачі ЕРН (15 генераторів).

Ітерація	**ОРЧ**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	38828,13	900507,0872	578,8923	135270,8007
250	38324,77	904139,9893	974,9242	147860,945
500	38283,26	871947,8408	740,9983	115983,2723
1000	39060,3	1644414,243	996,1076	1470911,701
1500	37922,94	912450,2997	579,6425	92550,81572
2000	38397,41	8895093,564	1360,6	23949012,87

Таблиця 5.8 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих АВЗ в залежності від максимальної кількості ітерацій для задачі ЕРН (15 генераторів).

Ітерація	**АВЗ**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	38272,53	143940443,4	1803,194	374931150,5
250	37342,1	85563074,91	985,1373	202876470,2
500	37688,68	50240064,73	886,0008	78372144,88
1000	37682,18	767515,1053	439,9639	215395,3803
1500	37646,23	725682,0965	505,5987	223144,6607
2000	37680,25	775457,5109	603,3749	311640,0875

Таблиця 5.9 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих модифікованим АВЗ в залежності від максимальної кількості ітерацій для задачі ЕРН (15 генераторів).

Ітерація	**Модифікація АВЗ**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	37001,32	57252859	597,1045	69604665
250	37831,88	3329955	387,4498	4081250
500	37972,31	2266401	312,5884	3052756
1000	37806,93	8609557	670,9717	18402733
1500	38192,28	778771,9	297,2419	122348,6
2000	38117,27	631825	313,3652	516718,4

За результатами першого експерименту для задачі ЕРН із системою для 15 генераторів найкращі результати математичного сподівання значень фітнес функції показав модифікований АВЗ для різної кількості ітерацій. Значення математичного сподівання фітнес функції алгоритму ОРЧ вказують, що він повільно виходить із локальних оптимумів. Генетичний алгоритм продемонстрував високі значення дисперсії для цільової функції, а це в свою чергу вказує на нестабільність розробленого алгоритму.

Проведемо подібний експеримент для задачі ЕРН для системи із 20 генераторів. Результати подані в таблицях 5.10-5.13.

Таблиця 5.10 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих ГА в залежності від максимальної кількості ітерацій для задачі ЕРН (20 генераторів).

Ітерація	**ГА**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	74639,96	74135369373	47326,38	28861067538
250	74748,79	73538941104	4652,167	33582616066
500	73371,69	77438954261	5646,831	31201062352
1000	73047,97	63146700491	3110,149	28560756240
1500	75906,41	60521730187	7922,178	20846134779
2000	75089,89	53806365064	4957,818	15577060071

Таблиця 5.11 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих алгоритмом ОРЧ в залежності від максимальної кількості ітерацій для задачі ЕРН (20 генераторів).

Ітерація	**ОРЧ**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	74347,91	50235368446	2054,647	12211298194
250	73341,7	66838941099	2501,427	31319168922
500	71678,12	68838954261	2002,914	26795978314
1000	72142,41	56646700534	2652,886	18003773360
1500	73411,67	54621730162	1729,336	1265753638
2000	73482,49	48306365038	1253,9	11731411239

Таблиця 5.12 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих АВЗ в залежності від максимальної кількості ітерацій для задачі ЕРН (20 генераторів).

Ітерація	**АВЗ**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	69504,65	73796171422	1030,304	14137493251
250	70079,48	66311454610	1008,402	12447692427
500	70425,87	61448558960	1513,14	13776698774
1000	70605,46	32004427504	1338,345	8647526127
1500	70586,89	15045093584	1301,533	10557490893
2000	79939,4	9023095049	1092,113	7985307701

Таблиця 5.13 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих МАВЗ в залежності від максимальної кількості ітерацій для задачі ЕРН (20 генераторів).

Ітерація	**МАВЗ**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	73076,7	76679180916	44866,85	887344496
250	76450,1	42881391512	60964,85	31022567
500	88107,8	34462711143	11168,15	2663155
1000	79249,08	12184589701	3444,485	504654
1500	80416,36	9665901470	4035,931	88227464
2000	81027,45	7208948606	6957,078	1260800

Аналізуючи значення математичного сподівання фітнес функції, генетичним та ОРЧ алгоритмам досить довго виходять із локальних оптимумів. Модифікований АВЗ та ГА мають велику розбіжність значень цільової функції, про що свідчить їхня дисперсія. Розроблений АВЗ показує низькі значення дисперсії для цільової функції. Найкращі результати для задачі ЕРН для системи із 20 генераторів показали алгоритми на основі методу вовчої зграї.

Розглянемо наступну задачу ДРН для системи із 15 генераторів. Результати наведені в таблицях 5.14-5.17.

Таблиця 5.14 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих ГА в залежності від максимальної кількості ітерацій для задачі ДРН (15 генераторів).

Ітерація	**ГА**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	34503	161124876864	10262	44467137651
250	38473	61420700225	282	12737674434
500	37969	86457491994	667	41236884610
1000	38206	90078313345	477	59210729000
1500	38148	67637886411	554	47611038487
2000	38700	86123780161	383	44847560812

Таблиця 5.15 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих алгоритмом ОРЧ в залежності від максимальної кількості ітерацій для задачі ДРН (15 генераторів).

Ітерація	**ОРЧ**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	37915	103124876899	753	43960259400
250	38482	62720700235	283	15845067116
500	37958	84157491986	653	36638396999
1000	38134	79178313553	397	43747612570
1500	38274	65637886462	530	42752009810
2000	38604	51123780290	370	38279191582

Таблиця 5.16 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих АВЗ в залежності від максимальної кількості ітерацій для задачі ДРН (15 генераторів).

Ітерація	**АВЗ**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	31340	1296141038794	250	326317256269
250	31624	903627119407	294	102656988391
500	31648	707801830906	204	178854738297
1000	31925	510808584432	158	155903159512
1500	31963	509777139266	158	122526157931
2000	32053	404016080913	227	36089144350

Таблиця 5.17 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих МАВЗ в залежності від максимальної кількості ітерацій для задачі ДРН (15 генераторів).

Ітерація	**МАВЗ**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	31300	1435420548901	207	211166045000
250	31624	1077992831380	194	253539075426
500	31737	860062117535	259	203405735964
1000	31953	531598774608	178	142449669203
1500	32069	476496057537	241	83375356250
2000	32141	454809408089	274	51716098866

Для задачі ДРН, що складається із 15 генераторів, всі розроблені алгоритми продемонстрували низькі значення дисперсії цільової функції. Класична реалізація алгоритму продемонстрували найкращі результати по всім критеріям. Особливо можна відмітити стійкість значень фітнес функції. Переглянувши результати фітнес функції, робимо висновок, що завдяки алгоритмам ОРЧ, АВЗ та МАВЗ можна отримати допустимі розв'язки для даної задачі за 1000 ітерацій.

Проведемо експерименти для останньої задачі – ДРН для системи із 20 генераторів. Результати наведені в таблицях 5.18-5.21.

Таблиця 5.18 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих ГА в залежності від максимальної кількості ітерацій для задачі ДРН (20 генераторів).

Ітерація	**ГА**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	76145,32	20124905837532	2137	8322037051732
250	79005,53	10435624469053	3532	3640101994434
500	79063,84	10702466968797	2132	2217694448901
1000	78771,24	10457517605557	1808	1503622819628
1500	79798,20	12221433865265	895	2242173750246
2000	80093,35	10411127807359	1066	2272821875246

Таблиця 5.19 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих алгоритмом ОРЧ в залежності від максимальної кількості ітерацій для задачі ДРН (20 генераторів).

Ітерація	**ОРЧ**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	75752,88	19924905837417	1406	3802078598883
250	78108,50	11585624469116	1395	1964657366287
500	78572,31	10502466968887	934	2157334313563
1000	79174,58	10527517605605	1516	1316287517275
1500	79849,84	12021434265303	932	2159750518352
2000	79701,40	10811127807389	1224	2571337033804

Таблиця 5.20 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих АВЗ в залежності від максимальної кількості ітерацій для задачі ДРН (20 генераторів).

Ітерація	**АВЗ**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	58767	59291229375904	479	4599640191291
250	59414	48087521614617	344	2826335848204
500	59681	38157405074977	678	2634811822867
1000	60236	31549795447330	555	2340088828701
1500	60922	28563449931016	474	2026175785068
2000	61035	25277311048867	497	1776116885851

Таблиця 5.21 – Порівняння математичного очікування та дисперсії вартості палива відносно значень цільової та фітнес функцій отриманих МАВЗ в залежності від максимальної кількості ітерацій для задачі ДРН (20 генераторів).

Ітерація	**МАВЗ**			
	Мат. сподівання		**Дисперсія значень**	
	Значень ц. ф.	**Значення ф. ф.**	**Значень ц. ф.**	**Значення ф. ф.**
100	58959	63104269088283	600	441307272456
250	59510	12968001932049	609	412515483661
500	59984	8679760532739	602	332326924397
1000	60401	7053559147150	324	291725171590
1500	62723	3167325735725	278	152741511099
2000	62838	2833281278438	188	197375407369

Вже після 500 ітерації модифікація АВЗ знаходить розв’язки, що задовольняють певним обмеженням задачі ДРН. Також варто відмітити стійкість класичної реалізації та модифікації АВЗ в порівнянні із ГА та ОРЧ.

Порівняємо час роботи даних алгоритмів для першої задачі – ЕРН для системи із 15генераторів (таблиця 5.22).

Таблиця 5.22 – Час роботи розроблених алгоритмів

Алгоритм	ГА	ОРЧ	АВЗ	МАВЗ
Час роботи алгоритму (мс)	3312,9	20959,1	1867,2	1874,2

У частині графічного матеріалу подані графіки залежності:

– фітнес функції від обраної максимальної кількості ітерацій для реалізованих алгоритмів;

– цільової функції від обраної максимальної кількості ітерацій для реалізованих алгоритмів.

Після проведення ряду експериментів над розробленими алгоритмами для різних задач можна підвести наступні висновки. ГА та ОРЧ досить швидко знаходять локальний оптимум і залишаються в ньому (при цьому високі значення фітнес-функції демонструють присутність значного порушення обмежень задачі). Також варто звернути увагу на довгий час роботи ГА. Натомість АВЗ та модифікація АВЗ демонструють тенденцію у поступовому спаданні значень фітнес-функції та знаходженні нових значень цільової функції на всіх етапах роботи. Для більшості задач розроблена модифікація АВЗ показує найкращі результати.

5.3 Дослідження алгоритму зведення розв’язку до допустимого

Для реалізованих алгоритмів застосовано алгоритм, що чвалається із процедур зведення розв’язку до допустимого. Проведено по одному тестовому запуску для кожного алгоритму для розв’язування задачі ДРН для системи із 15 генераторів. В таблиці 5.23 наведені результати цільової функції до та після зведення розв’язків.

Таблиця 5.23 – Порівняння значень цільової функції до та після роботи процедури зведення розв’язків до допустимих

Алгоритми	ГА	ОРЧ	АВЗ	МАВЗ
Результати роботи алгоритму без зведення розв’язку до допустимого	723745	722509	711811	709517
Результати роботи алгоритму із зведенням розв’язку до допустимого	762439	758230	740784	742263

Розроблена процедура перевіряє дотримання обмеження зміни швидкості робочої потужності та обмеження заборонених зон. При порушені одного із обмежень ми піднімаємо кількість виготовленої робочої потужності до допустимої. При цьому очевидно, що вартість палива буде рости, що показано у даній таблиці.

У таблиці 5.24 наведені значення фітнес функцій до та після роботи процедури зведення розв'язків до допустимих.

Таблиця 5.24 – Порівняння значень фітнес функції до та після роботи процедури зведення розв'язків до допустимих

Алгоритми	ГА	ОРЧ	АВЗ	МАВЗ
Результати роботи алгоритму без зведення розв'язку до допустимого	1257520431426	838847004416	1330703228022	4200840314711
Результати роботи алгоритму із зведенням розв'язку до допустимого	678104628189	518257870778	332588232698	2046413537824

Значення фітнес функції зменшились після завершення роботи процедури зведення розв'язку до допустимого. Це означає, що отримані розв'язки задовольняють всім обмеженням.

Також необхідно дослідити час роботи алгоритмів із використання процедури зведення розв'язку до допустимого. Для цього будемо обраховувати відсоток часу роботи процедури зведення відносно роботи основного алгоритму:

$$k = \frac{f_{пр.зв.}}{f_{ос.ал.}} \cdot 100\%,$$

де $f_{пр.зв.}$ – час роботи процедури зведення розв'язку до допустимого для даних, що були отримані в результаті роботи основного алгоритму, $f_{ос.ал.}$ – час роботи основного алгоритму. У таблиці 5.25 наведено порівняння часу роботи процедури відносно основного алгоритму.

Таблиця 5.25 – Порівняння часу роботи процедури зведення розв’язків до допустимих відносно основного алгоритму

Алгоритми	ГА	ОРЧ	АВЗ	МАВЗ
k	2,12%	2,01%	1,56%	1,68%

Час, який необхідний для процедури зведення розв’язку до допустимого, знаходиться в допустимих часових межах для процедур подібного виду.

Висновки до розділу

Експерименти проведені для порівняння ефективності роботи реалізованих алгоритмів та перевірки процедури зведення отриманих розв’язків до допустимих. Наведений процес проведення дослідження, що включає опис вхідних параметрів задач ЕРН та ДРН, параметри реалізованих алгоритмів, значення коефіцієнтів фітнес функції та характеристики апаратного забезпечення. Обрано по два типи задач ЕРН та ДРН із реальними даними. Проведено дослідження для визначення кількості «охоронців» у популяції для модифікації АВЗ.

Для кожної задачі виконано 100 тестових запусків кожного алгоритму для кожної кількості ітерації. Виконавши аналіз результатів, робимо висновок, що . ГА та ОРЧ досить швидко знаходять локальний оптимум і залишаються в ньому (при цьому високі значення фітнес-функції демонструють присутність значного порушення обмежень задачі). Для більшості задач розроблена модифікація АВЗ показує найкращі результати.

Також перевірено роботу процедури зведення розв’язку до допустимого. Отримані розв’язки задовольняють обмеженню зміни швидкості робочої потужності та обмеженню на заборонені зони за прийнятний час виконання.

ВИСНОВКИ

У магістерській дисертації досліджено задачу ефективного розподілу навантаження між електростанціями на прикладі задач ЕРН та ДРН, що описують процес розподілу робочої потужності між генераторами, враховуючи низку фізичні та експлуатаційні обмеження. Особливістю електроенергетичної промисловості є відсутність можливості зберігати виготовлену продукцію – вся невикористана електроенергія буде втрачена. Через це та високу вартість палива, необхідного для роботи електростанції, ефективний розподіл виробництва може призвести до суттєвої економії коштів.

У першому розділі проведено аналіз наукових робіт, що присвячені досліджуваній проблематиці. Розглянуто ряд підходів до розв’язування задач ЕРН та ДРН, а саме: лінійне програмування, нелінійне програмування, динамічне програмування, стохастичне програмування, методи мета евристичної оптимізації та гібридні технології. З огляду на особливості форми області значень цільової функції оптимізаційної задачі ЕРН чи ДРН, зазвичай перші три категорії методів швидко зупиняються в локальних оптимумах цільової функції, що далекі від глобальних. Крім того, вхідні та вихідні характеристики сучасних генераторів за своєю природою є нелінійними з різних причин, серед яких ефект різнорідного палива, нерівномірного навантаження вузлів (клапанів), обмеження швидкості зміни потужності та інші. Ці характеристики і призводять до виникнення цілого ряду локальних оптимумів, які створюють проблеми із знаходженням глобального оптимуму. Тому для розв’язування даних задач було обрано методи мета-евристичної оптимізації: ГА, ОРЧ та АВЗ.

У другому розділі наведено змістову постановку задачі ефективного розподілу навантаження. Також описано математичну постановку задачі для ЕРН, ЕРН без врахування втрат, ЕРН із точковим навантаженням на клапани та ДРН. Описано перехід від задачі із обмеженнями, що характеризують фізичні та експлуатаційні

характеристики електростанцій, до задачі без обмежень із використанням функцій штрафів.

У третьому розділі подані три алгоритми ройового інтелекту та генетичний алгоритм. На основі методу оптимізації вовчою зграєю була запропонована ідея оригінального АВЗ, що заключається в поділі зграї на «охоронців» та «мисливців». Також було розроблено алгоритм, який складається із процедур, для зведення отриманого розв’язку до допустимого.

У четвертому розділі описано засоби розробки, а саме опис обраної мови програмування, фреймворку, веб-сервера та СУБД. Наведено опис головних класів та їх схематичне зображення, яке наведено у додатку графічного матеріалу. Подано опис таблиць бази даних та вхідних і вихідних даних.

У п’ятому розділі описано процес проведення експериментів для порівняння ефективності роботи реалізованих алгоритмів та перевірки процедури зведення отриманих розв’язків до допустимих. Виконаний аналіз результатів дозволяє зробимо висновок, що ГА та ОРЧ досить швидко знаходять локальний оптимум і залишаються в ньому (при цьому високі значення фітнес-функції демонструють присутність значного порушення обмежень задачі). Для більшості задач розроблена модифікація АВЗ показала найкращі результати.

Розроблені алгоритми та програмні засоби можуть бути застосовані для розв’язування задач ефективного розподілу навантаження. Напрямком для подальших досліджень може бути розробка гібридних алгоритмів на основі розробленого оригінального АВЗ для розв’язування задач ЕРН та ДРН.

ПЕРЕЛІК ПОСИЛАНЬ

1 Бабич С. О. Алгоритми ройового інтелекту для задачі динамічного розподілу навантаження / С. О. Бабич, Л. Ф. Гуляницький // Науковий огляд. – 2018. - №3. – с. 32-47.

2 Бабич С. О. Модифікація алгоритму вовчої зграї для задачі динамічного розподілу навантаження / С. О. Бабич // Науковий огляд. – 2018. - №3. – с. 18-31.

3 Каплун В. В. Аналіз методів оптимізації мікроенергетичних систем (MicroGrid) на основі джерел розподіленої генерації / В. В. Каплун, О. П. Кравченко, В. В. Василенко, С. С. Макаревич, Р. В. Каплун // Вісник Київського національного університету технологій та дизайну. Серія : Технічні науки. - 2015. - № 2. - С. 5-17.с – Режим доступу: http://nbuv.gov.ua/UJRN/vknutdtn_2015_2_3

4 Wirfs-Brock Jordan. Lost In Transmission: How Much Electricity Disappears Between A Power Plant And Your Plug? [Електронний ресурс] / Jordan Wirfs-Brock // Inside Enegry. - 2015. Режим доступу: http://insideenergy.org/2015/11/06/lost-in-transmission-how-much-electricity-disappears-between-a-power-plant-and-your-plug.

5 Saad F. Optimal economic and environmental operation of electric power systems via modern meta-heuristic optimization algorithms / F. Saad. – 2011. – P. 1-3.

6 Delson J. K. Linear programming applications to power system economics, planning and operations // J. K. Delson, S. M. Shahidehpour // Power Systems, IEEE Transactions. – 1992. –V. 7. – №. 3. – P. 1155–1163.

7 Kurucz C. A linear programming model for reducing system peak through customer load control program / C. Kurucz, D. Brandt, S. Sim // Power Systems, IEEE Transactions. – 1996. – V. 11. – №. 4. –P. 1817–1824.

8 Khodr H. M. A linear programming methodology for the optimization of electric power-generation schemes / H. M. Khodr, J. F. Gomez, L. Barnique // Power Systems, IEEE Transactions. – 2002. –V. 17. – № 3. –P. 864–869.

9 Alguacil N. Transmission expansion planning: amixed-integer lp approach / N. Alguacil, A. L. Motto, A. J. Conejo // Power Systems, IEEE Transactions. – 2003. – V. 18. – №. 3. – P. 1070–1077.

10 Ahmad A. Optimization of Economic Load Dispatch Problem by Linear Programming Modified Methodology / A. Ahmad, A. Z. Khan // International Conference on Emerging Trends in Engineering and Technology. – London, 2014. – P. 76-79.

11 Ahmad A. Economic Load Dispatch Using Linear Programming: A Comparative Study / A. Ahmad, K. Hesham // International Journal of Applied Industrial Engineering. –2016. – V. 3(1).

12 Rahli M. Real power-system economic dispatch using a variable weights linear programming method / M. Rahli, L. Benasla, A. Belmadani, L. Abdelhakem-Koridak // Journal of Power Technologiesю – 2015. – V. 95 (1). – P. 34–39.

13 Liang R. H. A neural-based redispatch approach to dynamic generation allocation / R. H. Liang // IEEE Trans. Power Syst. – 1999. – V. 14. – № 4. – P.1388-1393.

14 Perez-Arriaga I. J. A nonlinear programming approach to optimal static generation expansion planning / I. J. Perez-Arriaga, J. Bogas // Power Systems, IEEE Transactions. – 1989. – V. 4. – № 3. – P. 1140–1146.

15 Gallego R. A. Optimal capacitor placement in radial distribution networks / R. A. Gallego, A. J. Monticelli, R. Romero // Power Systems, IEEE Transactions. – 2001. – V. 16. – № 4. – P. 630–637.

16 Nanda J. Economic emission dispatch with line flow constraints using a classical technique / J. Nanda, L. Hari, M.L. Kothari // IEE Proc Gener Trans Distrib. – 1994. – V. 141. – P. 10.

17 Chen C.L. Branch and bound scheduling for thermal generating units / C.L. Chen, S.C. Wang // IEEE Trans Energy Convers. – 1993. – V. 184. – P. 9.

18 Diaz-Dorado E. Optimal planning of unbalanced networks using dynamic programming optimization / E. Diaz-Dorado, J. C. Pidre // Power Systems, IEEE Transactions. –2004. – V. 19. – № 4. – P. 2077–2085.

19 Liang Z. X. A zoom feature for a dynamic programming solution to economic dispatch including transmission losses / Z. X. Liang, J. D. Glover // IEEE Trans. Power Syst. –1992. – V. 7. – № 2. – P. 544-550.

20 Li F. Hybrid genetic approaches to ramping rate constrained dynamic economic dispatch / F. Li, R. Morgan, D. William // Elect. Power Syst. Res. –1997. – V. 43. – P. 97-103.

21 Balamurugan R. An Improved Dynamic Programming Approach to Economic Power Dispatch with Generator Constraints and Transmission Losses / R. Balamurugan, S. Subramanian // Journal of Electrical Engineering & Technology. – 2008. – V. 3. – № 3. – P. 320-330.

22 Hansen P. A separable approximation dynamic programming algorithm for economic dispatch with transmission losses / Pierre Hansen, Nenad Mladenovi // Yugoslav Journal of Operations Research. – 2002. – P. 157-166.

23 Muro C. Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations / Muro C, Escobedo R, Spector L, Coppinger R // Behav Process. – 2011. – P. 2–7.

24 . Chiang C.L. Genetic-based algorithm for power economic load dispatch / C.L. Chiang // IEE Proc Gener Trans Distrib. – 2007. –V. 1. – P. 9.

25 Kannan S. Application and comparison of metaheuristic techniques to generation expansion planning problem / S. Kannan, S. M. R. Slochanal, N. P. Padhy // Power Systems, IEEE Transactions. – 2005 – V. 20. – № 1. – P. 466–475.

26 Li F. Towards more cost saving under stricter ramping rate constraints of dynamic economic dispatch problems-A Genetic based approach / F. Li, R. Morgan, D. Williams // Int. Conf. Genetic Algorithms Eng. Syst.: Innovations and Application. – 1997, Glasgow, UK. –P. 221-225.

27 Ongsakul W. Parallel micro genetic algorithm based on merit order loading solutions for constrained dynamic economic dispatch / W. Ongsakul, J. Tippayachai // Elect. Power Syst. Res. –2002. – V. 61. – № 2. – P.77-88.

28 Ma H. A genetic algorithm-based approach to economic dispatch of power systems / H. Ma, R.E. Smith // Engineering Mechanic. – P.212-214.

29 Rachna T. Genetic Algorithm for Solving the Economic Load Dispatch / T. Rachna // Electronics and Communication Engineering. –2014. – V. 7. – № 2. – P.523-528.

30 Basu M. Improved differential evolution for economic dispatch / M.Basu // International Journal of Electrical Power & Energy Systems. – 2014. - P. 855-861.

31 Balamurugan B. An improved differential evolution based dynamic economic dispatch with nonsmooth fuel cost function / B. Balamurugan, R. Subramanian // J. Electr. Syst. – 2003. – V. 3. – № 3. – P. 151-161.

32 Balamurugan B. Differential evolution-based dynamic economic dispatch of generating units with valve-point effects / B. Balamurugan, R. Subramanian // Elect. Power components Syst. – 2008. – V. 36. – № 1. – P. 828-843.

33 Perez-Guerrero R.E. Economic power dispatch with non-smooth cost functions using differential evolution / R.E. Perez-Guerrero, R.J. Cedenio-Maldonado // Proceedings of the 37th annual North American power symposium. – 2005. – P. 90.

34 Joned M. A. A. Solving dynamic economic dispatch using evolutionary programming / M. A. A. Joned, I. Musirin, T. K. Abdul Rahman // in Proc. 1st IEEE Int. Power Energy Conf. – 2006. – P. 144-149.

35 Swarup K. S. Constrained optimization using evolutionary programming for dynamic economic dispatch / K. S. Swarup, A. Natarrajan // in Proc. 3rd Int. Conf. Intell. Sensing Inform. Processing. – 2005. – P. 314-319.

36 Panigrahi C. K. Simulated annealing technique for dynamic economic dispatch / C. K. Panigrahi, P. K. Chattopadhyay, R. N. Chakrabarti, M. Basu // Elect. power components syst. – 2006. – V 34. – P. 577-586.

37 Hopfield J. J. Neurons with graded response have collective, computational properties like those of two-state neuron / J. J. Hopfield // Proc. Nat. Acad. Sci. –1984. – V. 81. – P. 3088- 3092.

38 Fukuyama Y. An application on neural network to dynamic dispatch using multi processors / Y. Fukuyama , Y. Ueki // IEEE Trans. Power Syst. – 1994. – V. 8. –№. 4. – P. 1299- 1307.

39 Liang R. H. A neural-based redispatch approach to dynamic generation allocation / R. H. Liang // IEEE Trans. Power Syst. – 2002. – V. 14. –№. 4. – P.. 1388-1393.

40 Гуляницький Л.Ф. Прикладні методи комбінаторної оптимізації: навч. посіб. / Л. Ф. Гуляницький, О. Ю. Мулеса // К. : Видавничо-поліграфічний центр «Київський університет», 2016. – 142 с.

41 Hassan R. A comparison of particle swarm optimization and the genetic algorithm / R. Hassan, B. Cohanim, O. De Weck, G. Venter // Proceedings of the 1st AIAA Multidisciplinary Design Optimization Specialist Conference. – 2005.

42 Arya L. Improved particle swarm optimization applied to reactive power reserve maximization / L. Titare, D. Kothari // International Journal of Electrical Power and Energy Systems. – 2010. – V. 32. – №. 5. –P. 368 – 374.

43 Coelho L.S. Solving economic load dispatch problems in power systems using chaotic and Gaussian particle swarm optimization approaches / L.S. Coelho, C. Lee // Int. J. Elec. Power. – 2008. – P. 297-307.

44 Kumar Nimish. Economic load dispatch using improved particle swarm optimization algorithms / Nimish Kumar, Uma Nangia, Kishan Bhushan Sahay // Power India International Conferenceю – 2014. – V. 6.

45 Dorigo M. Optimization, Learning and Natural Algorithms / M. Dorigo // Politecnico di Milano. – 1992.

46 Aristidis V. An ant colony optimization (ACO) algorithm solution to economic load dispatch (ELD) problem / Vlachos Aristidis // Department of Informatics, University of Piraeus. – P. 153-160.

47 Swarup K. S. Ant Colony Optimization for Economic Generator Scheduling and Load Dispatch / K. S. Swarup // Proceedings of the 6th WSEAS Int. Conf. on EVOLUTIONARY COMPUTING. – 2005. Lisbon, Portugal. –V.16. – P.167-175.

48 Mirjalili S. Grey Wolf Optimizer / S. Mirjalili, S. M. Mirjalili, A. Lewis // Advances in Engineering Software. – 2014. – V. 69. – P. 46-61.

49 Mirjalili S. How effective is the Grey Wolf optimizer in training multi-layer perceptrons / S. Mirjalili // The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies. – 2015. – № 43. – P. 645.

50 Sharma Sudhir. Economic Load Dispatch Using Grey Wolf Optimization / Sudhir Sharma, Shivani Mehta, Nitish Chopra // Journal of Engineering Research and Applications. – 2015. – P. 128-132

51 Pradhan M. Grey Wolf Optimization applied to economic load dispatch problems / M. Pradhan, P. Kumar Roy, T. Paul // Electrical Power and Energy Systems. – 2016. – V. 83. – P. 325-334.

52 Prashar S. Formulation of improved grey wolf optimization methodology for EELD problem / S. Prashar // International Journal of Science Technology and Engineering. – 2016. – V.4. – № 5. – P. 23-31.

53 Attaviriyanupap D. SQP for dynamic economic dispatch with nonsmooth incremental fuel cost function / D. Attaviriyanupap, H. Kita, E. Tanaka, J. Hasegawa // IEEE Trans. Power Syst. – 2002. – V. 17. – № 2. –P. 411-416.

54 A hybrid HNN-QP approach for dynamic economic dispatch problem / Y. Abdelaziz, M. Z. Kamh, S. F. Mekhamer, M. A. L. Badr // Elect. Power Syst. Res. – 2006. – V.4. – № 5. – P. 23-27.

55 Ongsakul W. Constrained dynamic economic dispatch by simulated annealing/genetic algorithms / W. Ongsakul, N. Ruangpayoongsak // Proc. 22nd Power Ind. Comput. Applicat. – 2001.Sydney, Australia. –P. 207-212.

56 Happ H. H. Optimal power dispatch – A comprehensive survey / H. H. Happ// IEEE Trans. Power Apparatus Syst. – 1977. – P. 841-854.

57 Kwatny H. G. On the structure of optimal area controls in electric power networks / H. G. Kwatny, T. E. Bechert // IEEE Trans. Autom. Control. – 1973. – V. 18. – P. 176-172.

58 Han X. S. Effective economic dispatch model and algorithm / X. S. Han, H. B. Gooi // Elect. Power Energy Syst. – 2007. – V. 29. – P.113-120.

59 Attaviriyanupap P. A fuzzy-optimization approach to dynamic economic dispatch considering uncertainties / P. Attaviriyanupap, H. Kita, E. Tanaka, J. Hasegawa // IEEE Trans. Power Syst. – 2004. – V. 19. – № 3. – P. 1299- 1307.

60 Lin W. M. Bid-based dynamic economic dispatch with an efficient interior point algorithm / W. M. Lin, S. J. Chen // Elect. Power Energy Syst. – 2002. – V. 24. – P. 51-57.

61 Li F. Fast and accurate power dispatch using a relaxed genetic algorithm and a local gradient technique / F. Li, R. K. Aggarwal // Expert Syst. Applicat. –2000. –V. 19. – P. 159-165.

62 Victoire T. A. A. A modified hybrid EP-SQP approach for dynamic dispatch with valve-point effect / T. A. A. Victoire, A. E. Jeyakumar // Int. J. Elect. Power Energy Syst. – 2005. –V. 27. – № 8. –P. 594-601.

63 Victoire T. A. Deterministically guided PSO for dynamic dispatch considering valve-point effect / T. A. A. Victoire, A. E. Jeyakumar // Electr. Power Syst. Res. – 2005. –V. 73. – № 3. –P. 313- 322.

64 Victoire T. A. Reserve constrained dynamic dispatch of units with valve-point effects / T. A. A. Victoire, A. E. Jeyakumar // IEEE Trans. Power Syst. – 2005. –V. 20. – № 3. –P. 1273- 1282.

65 Yuan X. A hybrid differential evolution method for dynamic economic dispatch with valve-point effects / X. Yuan, L. Wang, Y. Zhang, Y. Yuan // Expert Syst. Appl. – 2008.

66 Zhang G.-L. Dynamic economic load dispatch using hybrid genetic algorithm and the method of fuzzy number ranking / G.-L. Zhang, H.-Y. Lu, G.-Y. Li, G.-Q. Zhang // Proc. 4th Int. Conf. Mach. Learning Cybernetics. – 2005. –P.2472-2477.

67 Zhang G.-L. A new hybrid real-coded genetic algorithm and application in dynamic economic dispatch / G.-L. Zhang, H.-Y. Lu, G.-Y. Li, H. Xie // Proc. 6th World Congr. Intell. Control and Autom. – 2006. Dalian, China. – P. 3627- 3632.

68 Titus S. A hybrid EP-PSO-SQP algorithm for dynamic dispatch considering prohibited operating zones / S. Titus, A. E. Jeyakumar // Elect. Power Components Syst. – 2008. – V. 36. –P. 449-467.

69 Rahmat N.A. Hybrid differential evolution-ant colony optimization for economic load dispatch problem / N.A. Rahmat, I. Musirin // Journal of Theoretical and Applied Information Technology. – 2013. – P. 680-690.

70 Флэнаган Д. Язык программирования Ruby [Текст]/ Д. Флэнаган, Ю. Мацумото // СП.: Питер 2011. – 496с.

71 Сайт групи розробників мови програмуванн Ruby [Електронний ресурс] // Режим доступу: https://www.ruby-lang.org

72 Сайт групи розробників фреймворку Ruby On Rails [Електронний ресурс] // Режим доступу: http://www.rubyonrails.ru/

73 Сайт розробника сервера Nginx [Електронний ресурс] // Режим доступу: http://nginx.org/

74 November 2014 Web Server Survey [Електронний ресурс] // Режим доступу: http://news.netcraft.com/archives/2014/11/19/november-2014-web-server-survey.html

75 Usage of web servers broken down by ranking [Електронний ресурс] // Режим доступу: http://w3techs.com/technologies/cross/web_server/ranking

76 Сайт розробників сервера Unicorn [Електронний ресурс] // Режим доступу: http://unicorn.bogomips.org/

77 Chelimsky D. The RSpec Book: Behaviour Driven Development with RSpec, Cucumber, and Friends [Текст]/ D. Chelimsky, D. Astels, B. Helmkamp, D. North // F.: Pragmatic Bookshelf. – 2010. – 448c.

ПЛАКАТ 1
РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНИХ ДОСЛІДЖЕНЬ

ПЛАКАТ 2
ДІАГРАМА КЛАСІВ

ПЛАКАТ 3
ДІАГРАМА ПАКЕТІВ

ПЛАКАТ 4
ДІАГРАМА ПОСЛІДОВНОСТІ

ПЛАКАТ 5
БЛОК-СХЕМИ РОЗРОБЛЕНИХ АЛГОРИТМІВ: ОРЧ, АВЗ ТА МОДИФІКАЦІЯ АВЗ

ПЛАКАТ 6

БЛОК-СХЕМИ ПРОЦЕДУР ЗВЕДЕННЯ РОЗВ’ЯЗКУ ДО ДОПУСТИМОГО

ПЛАКАТ 7
ЕКРАННІ ФОРМИ